

Vysoká škola báňská - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Simulátor poruchových stavů Automotive senzorů
Failure State Simulator of Automotive Sensors

Zadání diplomové práce

Student: **Bc. Tomáš Skotnica**
Studijní program: N2649 Elektrotechnika
Studijní obor: 2612T041 Řídicí a informační systémy
Téma: **Simulátor poruchových stavů Automotive senzorů**
Failure State Simulator of Automotive Sensors
Jazyk vypracování: čeština

Zásady pro vypracování:

1. Rozbor problematiky rozhraní SENT a popis sběrnic užívaných v Continental Automotive Czech Republic.
2. Rozbor možných chybových stavů senzorů. (chyby v datech, chyby na úrovni fyzické vrstvy)
3. Návrh HW a realizace funkčního vzorku.
4. Realizace a testování zařízení.
5. Popis možností komunikace s PC.
6. Zhodnocení výsledků, závěr.

Seznam doporučené odborné literatury:

- [1] SAE J2716. *SENT - Single Edge Nibble Transmission for Automotive Applications*. Warrendale: SAE International, 2016.
- [2] ISO 11898. *Road vehicles - Controller area network (CAN)*. Switzerland: International Organization for Standardization, 2015. p.65.
- [3] ISO 17987. *Road vehicles - Local Interconnect Network (LIN)*. Switzerland: International Organization for Standardization, 2016.
- [4] ISO 10605. *Road vehicles - Test methods for electrical disturbances from electrostatic discharge*. Switzerland: International Organization for Standardization, 2008. p.50. ICS : 43.040.10.
- [5] SKOTNICA, Tomáš. *Absolvování individuální odborné praxe*. Ostrava, 2017. Bakalářská práce. Vysoká škola báňská - Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, Katedra kybernetiky a biomedicínského inženýrství.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Jan Žídek, CSc.**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 30.04.2019



.....
Bc. Tomáš Skotnica

Prohlášení zástupce spolupracující právnické nebo fyzické osoby

Continental Powertrain Czech Republic s.r.o.

Na rovince 879

72000 Ostrava-Hrabová

Česká republika

Souhlasím se zveřejněním této diplomové práce „Simulátor poruchových stavů Automotive senzorů“ dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.



Continental Powertrain Czech Republic s.r.o.
Na Rovince 879, 720 00, Ostrava-Hrabová
IČ: 07574622 DIČ: CZ07574622

A handwritten signature in blue ink, appearing to read "Miroslav Lepš".

V Ostravě dne 30.04.2019

.....
Continental Powertrain Czech Republic s.r.o.

Poděkování

Tímto bych chtěl poděkovat zaměstnancům společnosti Continental Powertrain Czech Republic s.r.o., za cenné rady a vedení během zpracovávání této diplomové práce.

Dále bych chtěl poděkovat svému vedoucímu práce ze strany VŠB-TU Ostrava doc. Ing. Janu Žídkovi, CSc. a v neposlední řadě také své rodině a blízkým za veškerou podporu.

Abstrakt

Tématem této diplomové práce je návrh a realizace simulátoru poruchových stavů teplotních senzorů s rozhraním SENT pro společnost Continental Powertrain Czech Republic s.r.o.

Realizace simulátoru poruchových stavů senzorů je provedena ve formě přenosného zařízení s mikrokontrolérem. Pro realizaci FW simulátoru je využit jazyk C.

Součástí práce je rozbor základní problematiky elektronických systémů určených pro automobilový průmysl. Práce se dále zabývá rozbohem vybraných sběrnic a rozhraní využívaných v automobilovém průmyslu pro senzorické systémy. Z hlediska senzorických systémů je uveden rozbor možných poruchových stavů s návazností na testování v oblasti EMC.

Klíčová slova

SENT, automobilový průmysl, MCU, C, FW, HW, simulace, testování

Abstract

Theme of this diploma thesis is design and realisation of failure state simulator of temperature sensors with SENT interface for company Continental Powertrain Czech Republic s.r.o.

The simulator is realised as portable device with microcontroller. C language is used for FW implementation.

Part of the thesis is basic analysis of electronic systems problematics designated for automotive industry. Thesis is also deals with analysis of chosen communication buses and interfaces for automotive industry. From the side of sensory systems an analysis is given of the failure states and its dependency to testing in area of EMC.

Keywords

SENT, automotive industry, MCU, C, FW, HW, simulation, testing

Obsah

Seznam ilustrací	12
Seznam tabulek	13
Seznam použitých zkratk a symbolů	14
1 Úvod	16
2 Sběrnice a rozhraní v automobilovém průmyslu	17
2.1 Elektronika v automobilovém průmyslu	17
2.2 Obecný popis komunikací	17
2.3 SENT	18
2.3.1 Princip funkce komunikačního rozhraní	18
2.3.2 Struktura datového rámce - rychlý kanál	19
2.3.2.1 Synchronizace	20
2.3.2.2 Spolehlivost kontrolního součtu	20
2.3.3 Struktura datového rámce - pomalý kanál	21
2.3.3.1 Krátká pomalá zpráva	21
2.3.3.2 Rozšířená pomalá zpráva	22
2.3.4 Multiplexování rychlého kanálu	23
2.3.5 Fyzická vrstva	24
2.4 CAN	25
2.4.1 Datové rámce	26
2.4.2 Arbitráž na sběrnici	26
2.4.3 Acknowledgement - Potvrzování komunikace	26
2.4.4 Bit stuffing na sběrnici	27
2.4.5 Detekce chyb	27
2.5 LIN	27
2.5.1 Složení zařízení na LIN sběrnici	28
2.5.2 Komunikační rámce	28
2.5.3 Časová determinističnost	29
2.5.4 Kontrolní součet	29
2.6 PSI5	30
2.6.1 Komunikace senzoru s ECU	31
2.6.2 Komunikace ECU se senzorem	31

2.6.3	Topologie propojení senzorů s ECU	31
2.6.4	Asynchronní mód	32
2.6.5	Synchronní (sběrníkový) mód	32
2.6.6	Daisy-chain mód	33
2.7	Analogová rozhraní	34
2.8	Srovnání vybraných sběrnic a rozhraní	34
3	Poruchové stavy	36
3.1	EMC a možné vlivy na poruchové stavy	36
3.1.1	Obecné informace	36
3.1.2	Přechodové jevy	36
3.1.3	Rušivé signály na vedení	36
3.2	Obecné informace o poruchových stavech	37
3.3	Rozdělení poruchových stavů senzorů v automobilovém průmyslu	37
3.3.1	Stavová hlášení senzoru	37
3.3.2	Chyby v datech	37
3.3.3	Chyby komunikačního protokolu	38
3.3.4	Chyby fyzické vrstvy	38
4	Návrh a realizace zařízení	39
4.1	Požadavky na zařízení	39
4.2	Návrh HW	39
4.2.1	SENT výstup	40
4.2.1.1	Push-Pull výstup	40
4.2.1.2	DAC + Buffer	43
4.2.2	Mikrokontrolér	44
4.2.3	USB	44
4.2.4	Ovládací a zobrazovací prvky	45
4.2.5	Napájecí zdroj	46
4.2.6	Realizovaný simulátor	48
4.3	Návrh FW	49
4.3.1	Návrh architektury FW	49
4.3.2	SENT protokol	50
4.3.3	GUI	52
4.3.4	Ovládací prvky	55
4.3.5	Grafický displej	56
4.3.6	UART	56

5	Možnosti komunikace s PC	58
5.1	Návrh komunikačního protokolu	58
6	Testování vyvinutého zařízení	60
6.1	Popis možného senzoru	60
6.2	Verifikace funkčnosti SENT rozhraní	61
6.3	Generování chyb na rozhraní SENT	63
7	Zhodnocení výsledků a závěr	68
	Literatura	70
	Seznam příloh	71

Seznam ilustrací

1	Příklad možného rámce přenášeného přes rozhraní SENT[6]	20
2	Příklad možné implementace rozhraní SENT[6]	25
3	CAN Classic Base rámec [1]	26
4	Rozložení master a slave úloh v zařízeních na sběrnici [3]	28
5	Vysílání Headeru a odpovědi [3]	28
6	Header a datový rámec [3]	29
7	Členové PSI5 organizace [5]	30
8	Možná propojení ECU a senzorů [4]	32
9	Propojení ECU se senzorem v asynchroním režimu a časový diagram komunikace [4]	32
10	Propojení ECU se senzory v synchroním režimu a časový diagram komunikace [4]	33
11	Propojení ECU se senzory v daisy-chain režimu a časový diagram komunikace [4]	33
12	Blokový diagram navrženého zařízení	39
13	Návrh push-pull výstupního obvodu pro SENT	40
14	Simulace tvarovacího π -článku	42
15	Reálný měřený průběh	42
16	Návrh DAC výstupního obvodu pro SENT	43
17	Mikrokontrolér s okolními komponenty pro zajištění funkčnosti	44
18	USB - vstupní ochrana proti ESD	45
19	USB - UART převodník	45
20	Displej s převodníkem logických úrovní z 5 V na 3,3 V	46
21	Měnič z 7 - 28 V na 5 V pro napájení celého zařízení	47
22	Lineární regulátor z 5V na 3,3V pro logiku LCD	47
23	Fotka realizovaného simulátoru - strana TOP	48
24	Fotka realizovaného simulátoru - strana BOT	49
25	Vývojový diagram se základním rozložením architektury FW	50
26	Vývojový diagram pro rutinu přerušování zajišťující generování SENT protokolu	51
27	Vývojový diagram pro rutinu přerušování zajišťující přípravu zpráv pro odeslání	52
28	Grafické uživatelské rozhraní - úvodní obrazovka	53
29	Grafické uživatelské rozhraní - nastavení teplot	53
30	Grafické uživatelské rozhraní - nastavení časů	53
31	Grafické uživatelské rozhraní - nastavení FMI kódů, chyby CRC a chyby S&C nibblu	54
32	Grafické uživatelské rozhraní - nastavení FMI kódů, chyby CRC a chyby S&C nibblu - aktivní chyby	54
33	Grafické uživatelské rozhraní - nastavení simulace opakování chyb	54

34	Vývojový diagram pro rutinu přerušení zajišťující detekci stisku tlačítek	55
35	Vývojový diagram pro rutinu přerušení zajišťující obsluhu vysílání na UART	57
36	Vývojový diagram pro rutinu přerušení zajišťující obsluhu příjmu na UART	57
37	Dekódované rámce pomocí osciloskopu, aktivní multiplexování rychlého kanálu (FC, DCC)	62
38	Dekódovaný rámec pomocí osciloskopu, aktivní multiplexování rychlého kanálu (FC, DCC)	62
39	Simulace tří souvislých chybových rámců	65
40	Simulace čtyř nesouvislých chybových rámců	66
41	Simulace chyby CRC rychlého rámce	66
42	Simulace chyby S&C nibblu -> chyba pomalého rámce	67

Seznam tabulek

1	Nároky na komunikační rozhraní či sběrnici dle oblasti využití	18
2	Příklady doba trvání zpráv pro různé hodnoty tick time	19
3	Příklady možného rozložení hodnot v datech[6]	20
4	Nibble Status a Komunikace[6]	21
5	Skladba krátké pomalé zprávy[6]	22
6	Skladba rozšířené pomalé zprávy[6]	22
7	Příklad ID pomalých zpráv (8-bit ID)[6]	23
8	Příklad pro teplotní senzor se 4 termočlánky, 16 bitové hodnoty	24
9	Porovnání vybraných sběrnic a rozhraní	35
10	Parametry definované normou[6]	41
11	Výsledky simulace a reálného měření	41
12	Definice symbolů SOF, EOF, escape sekvence	58
13	Typy příkazů v navrženém komunikačním protokolu	58
14	Příklady jednotlivých rámců	59
15	Specifikace senzoru \$048	60
16	Specifikace navrženého senzoru pro účely této diplomové práce - část 1	61
17	Specifikace navrženého senzoru pro účely této diplomové práce - část 2	61
18	Specifikace navrženého senzoru pro účely této diplomové práce - část 3	61
19	Dekódované rámce vyslané simulátor konfigurovaným dle specifikace senzoru v kapitole 6.1	63
20	Verifikace modifikovaných hodnot tick time, 12 μ s, 6 μ s a 3 μ s	64

Seznam použitých zkratk a symbolů

Zkratka	Zkrácený výraz	Vysvětlení
ABS	Anti-lock Brake System	systém proti zablokování brzd
ASCII	American Standard Code for Information Interchange	kódovací tabulky s definicí znaků anglické abecedy
ASIC	Application Specific Integrated Circuit	integrováný obvod vyvinutý pro konkrétní aplikaci
CAN	Controller Area Network	sběrnice CAN
COM	Communication Port	sériový port (UART, RS232)
CRC	Cyclic Redundancy Check	cyklický redundantní součet
DAC	Digital-to-Analog Converter	digitálně analogový převodník
DCC	Data Consistency Check	speciální nibble pro multiplexování rychlého kanálu SENT rozhraní
ECU	Electronic Control Unit	řídící jednotka
EMC	Electromagnetic compatibility	elektromagnetická kompatibilita
EMS	Electromagnetic Susceptibility	elektromagnetická odolnost
ESP	Electronic Stability Program	elektronický stabilizační program
FC	Frame Control	speciální nibble pro multiplexování rychlého kanálu SENT rozhraní
FMI	Failure Mode Indicator	indikátor poruchového stavu
FW	Firmware	
GND	Ground	
GPIO	General Purpose Input Output	univerzální vstupně výstupní pin/port
HW	Hardware	
IOC	Interrupt On Change	Periferie vyvolávající přerušení při změně logické úrovně
LDO	Low Dropout Stabilizator	stabilizátor s nízkým úbytkem napětí
LIN	Local Interconnect Network	sběrnice LIN
LISN	Line Impedance Stabilization Network	umělá síť
LSB	Least Significant Byte/Bit	nejméně významný bajt/bit
LSN	Least Significant Nibble	nejméně významný nibble
MCU	Microcontroller unit	mikrokontrolér
MSB	Most Significant Byte/Bit	nejvíce významný bajt/bit
MSN	Most Significant Nibble	nejvíce významný nibble

Zkratka	Zkrácený výraz	Vysvětlení
NRZ	Non-Return-to-Zero	kódování, kdy je bit dán po celou jeho délku jedním logickým stavem
PLL	Phase Locked Loop	fázový závěs
PSI5	Peripheral Sensor Interface	sběrnice/rozhraní PSI5
PWM	Pulse Width Modulation	pulsně šířková modulace
RAM	Random Access Memory	paměť s náhodným přístupem
RE	Radiated Emissions	vyzařované rušení
RI	Radiated Immunity	odolnost proti vyzařovanému rušení
SENT	Single Edged Nibble Transmission	rozhraní SENT
SPI	Serial Peripheral Interface	rozhraní SPI
S&C	Status & Communication nibble	část rychlé zprávy u SENT rozhraní
TVS diode	Transient Voltage Suppresion diode	transil, ochrana proti napětovým špičkám
V_{CC}		kladné napájecí napětí

1 Úvod

Cílem této práce je návrh a realizace simulátoru poruchových stavů senzorů s rozhraním SENT. Zadání a požadavky na tuto diplomovou práci vznikly ve spolupráci se společností Continental Powertrain Czech Republic s.r.o.

Tato diplomová práce z části navazuje na mou bakalářskou práci, kterou jsem vykonával formou odborné praxe ve společnosti Continental Powertrain Czech Republic s.r.o.. Během praxe jsem pracoval na vývoji obdobného zařízení, které bylo určeno pro simulování senzorů na sběrnici CAN. Vzhledem k naprosto odlišnému komunikačnímu protokolu a dalším omezením, které by vznikly rozšířením původního simulátoru bylo rozhodnuto o vývoji nového zařízení.

Motivací pro vývoj zařízení, které je v této práci popsáno je potřeba verifikace funkčnosti testovacích zařízení během jejich vývoje. V rámci testování senzorů se využívá řada vlastních testovacích zařízení. Taková zařízení je nutné během jejich vývoje testovat, což může být v případě některých typů chyb značný problém z důvodu komplikovanosti simulace těchto chyb na reálném senzoru. Z toho důvodu byl požadavek na vývoj simulátoru poruchových stavů senzorů s rozhraním SENT.

Diplomová práce je rozdělena na teoretickou a praktickou část.

Teoretická část práce obsahuje popis komunikačních sběrnic a rozhraní využívaných v automobilovém průmyslu, součástí teoretické části diplomové práce je dále obecný rozbor specifik a požadavků kladených na elektroniku využívanou v automobilovém průmyslu. V další části diplomové práce je rozbor možných poruchových stavů, se kterými je možné se setkat. Přímému rozboru poruchových stavů předchází popis problematiky elektromagnetické kompatibility (EMC) s výběrem některých testů, kterým jsou senzory podrobovány, které mají za úkol odhalit poruchové stavy či testovat imunitu senzoru na vlivy rušení, které může poruchové stavy způsobovat.

Praktická část diplomové práce je dále členěna na následující části. První část rozebírá požadavky a specifikace, které jsou na zařízení kladeny. V následujících kapitolách je popsán návrh HW a FW simulátoru. Následuje kapitola zabývající se možnostmi komunikace zařízení s PC. V poslední kapitole praktické části diplomové práce jsou zpracována data z testování simulátoru. Z důvodu utajení vyvíjených produktů společnosti Continental Powertrain Czech Republic s.r.o. je pro účely této práce vytvořena vlastní specifikace senzoru.

2 Sběrnice a rozhraní v automobilovém průmyslu

2.1 Elektronika v automobilovém průmyslu

Moderní automobil je vysoce komplexní systém, ve kterém se s každou novou generací rozšiřuje množství elektroniky. Velká část dříve čistě mechanických systému byla rozšířena, či úplně nahrazena elektronickými systémy.

Elektronika automobilu se skládá z celé řady řídicích jednotek, které zajišťují širokou funkcionalitu od řízení spalovacího agregátu, přes bezpečnostní prvky (airbag, ABS, ESP, atd.) až po prvky zvyšující komfort pasažérů (nastavitelná zpětná zrcátka, polohování sedadel, navigace, multimediální systémy). Pro zajištění žádané funkcionality využívají řídicí jednotky velké množství různých senzorů a aktuátorů.

Na elektroniku využívanou v automobilovém průmyslu je kladena řada specifických nároků. Z jedné strany působí tlak ekonomiky trhu, která se snaží snižovat cenu elektroniky. Z opačné strany působí tlak na velice vysokou spolehlivost a bezpečnost systémů. Jedná se o silně si konkurující požadavky, které není většinou možné úplně snadno sloučit.

Obecně lze automobilovou elektroniku rozdělit do následujících kategorií:

- Chassis & safety - brzdy, světla, ABS, ESP, tlak v pneumatikách, ...
- Powertrain - řízení spalovacího agregátu, filtrace zplodin, převodovka, ...
- Body & comfort - kliky dveří, zrcátka, topení, klimatizace, ...
- Infotainment & Telematics - přístrojový panel, rádio, navigace, ...

2.2 Obecný popis komunikací

V předchozí kapitole byla představena kategorizace elektroniky využívané v automobilech. Pro zajištění provozuschopnosti automobilu je nutné, aby řídicí jednotky dokázaly komunikovat se svými senzory a aktuátory. Samozřejmě je nutná i komunikace řídicích jednotek navzájem mezi sebou. K tomuto účelu se využívají komunikační rozhraní a sběrnice.

V případě automobilu se využívají rozhraní a sběrnice přímo určené pro automobilový průmysl, jelikož tyto sběrnice jsou navrženy a optimalizovány pro provoz v automobilu. Konkrétní výběr sběrnice či rozhraní vždy záleží na kritičnosti daného systému a požadované datové propustnosti. V tabulce 1 je základní přehled systémů s přehledem obecných požadavků na komunikační rozhraní či sběrnici.

Tab. 1: Nároky na komunikační rozhraní či sběrnici dle oblasti využití

	Odolnost	Determinističnost	Datová propustnost	Zabezpečení
Chassis & Safety	Vysoká	Vysoká	Nízká/Střední	Ne
Powertrain	Vysoká	Vysoká	Nízká/Střední	Ne
Body & Comfort	Nízká	Nízká	Nízká/Střední	Ne
Infotainment & Telematics	Nízká	Nízká	Vysoká	Ano

2.3 SENT

„Single Edged Nibble Transmission - SENT“ je jedním z rozhraní často aplikovaných v automobilovém průmyslu pro komunikaci periferních senzorů s řídicí jednotkou automobilu. Rozhraní SENT našlo své uplatnění díky své jednoduchosti na implementaci, dostatečné přenosové rychlosti a minimálním implementačním nákladům.

Komunikační rozhraní SENT je definováno normou SAE J2716. Norma definuje poměrně zásadní část přenosového protokolu a ponechává relativně nízký stupeň volnosti pro implementaci změn.

Základní technické parametry:

- třívodičové připojení senzoru - Vcc, GND, SENT
- datová linka na 5 V logice
- přenosová rychlost dle hodnoty „tick time“ a obsahu dat v rámci
- 1 až 6 nibblů v rámci (nibble - polovina bajtu, respektive čtyři bity)
- jedná se o rozhraní, tedy komunikace pouze dvou zařízení

2.3.1 Princip funkce komunikačního rozhraní

Ve svém principu se jedná o značně atypickou komunikaci. Přenos dat se provádí po již zmíněných nibblech.

Dalším specifikem tohoto typu komunikace je to, že se mění přenosová rychlost v závislosti na přenašených datech. K tomu dochází jelikož hodnota v nibblu je reprezentována délkou jeho trvání.

Časové parametry:

- perioda „tick time“ v rozsahu 3 až 90 $\mu\text{s} \pm 20\%$
- doba trvání nibblu 12 až 27 „ticků“

- minimální doba trvání rámce 154 až 270 „ticků“

Doba trvání nibblu je dána vztahem $T = 12 \cdot ticktime + x \cdot ticktime$, kde $x \in \langle 0, 15 \rangle$.

Tab. 2: Příklady doba trvání zpráv pro různé hodnoty tick time

Tick time μs	Minimální délka μs	Maximální délka μs	Rozsah datové propustnosti kbit/s	
3	$462 \pm 20\%$	$810 \pm 20\%$	$\approx 24,7$	$\approx 64,9$
45	$6930 \pm 20\%$	$12150 \pm 20\%$	$\approx 1,6$	$\approx 4,3$
90	$13860 \pm 20\%$	$24300 \pm 20\%$	$\approx 0,8$	$\approx 2,2$

2.3.2 Struktura datového rámce - rychlý kanál

Typicky se využívá pro přenos měřených dat ze senzorů. Rychlý rámec se skládá z následujících částí.

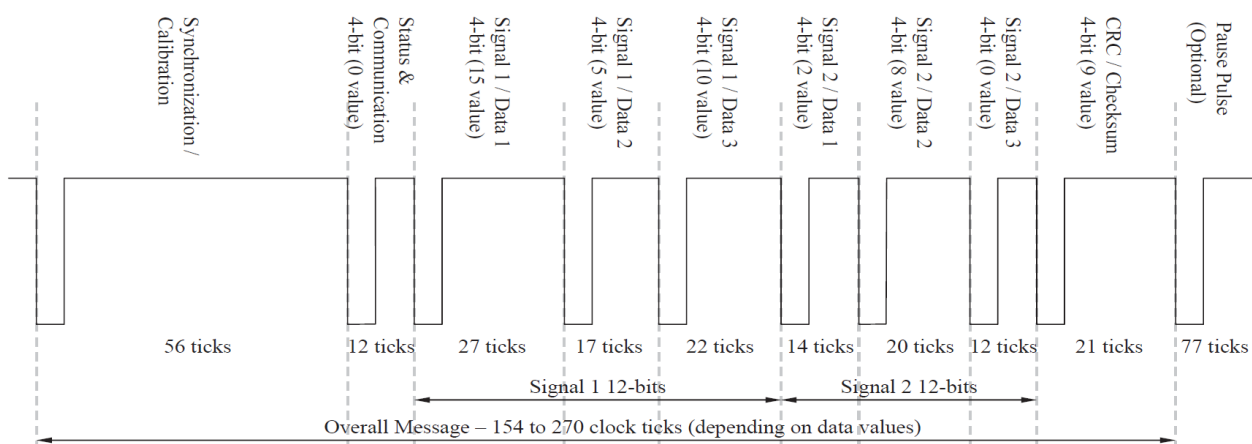
Synchronizační/kalibrační část - jedná se o část zprávy, která zajišťuje synchronizaci odchylek mezi taktem vysílače a přijímače (z dlouhodobého hlediska $\pm 20\%$). Celková délka vždy 56 „ticků“.

Status a Komunikace (S&C) - jedná se o nibble tedy 12 až 27 „ticků“. Obsahuje 2 bity pomalého kanálu a 2 signalizační bity dle aplikace.

Datové nibbly - 1 až 6 nibblů obsahující data či signalizaci (např. dle tabulky 3, každý 12 až 27 „ticků“).

CRC/kontrolní součet - dvě možné implementace aktuální doporučená a původní pro zajištění kompatibility.

Pauza - volitelně je možné přidávat pauzu, například pro zajištění pevné doby mezi začátky vysílání jednotlivých zpráv.



Obr. 1: Příklad možného rámce přenášeného přes rozhraní SENT[6]

Tab. 3: Příklady možného rozložení hodnot v datech[6]

Data / signalizace	16-bit hodnoty	14-bit hodnoty	12-bit hodnoty	10-bit hodnoty	8-bit hodnoty
Indikace chyb / speciální zprávy	65529	16377	4089	1017	249

	65535	16383	4095	1023	255
Měřená data	1	1	1	1	1

	65528	16376	4088	1016	248
Inicializace	0	0	0	0	0

2.3.2.1 Synchronizace

Každý rychlý rámec začíná synchronizačním pulsem, ECU porovnává délky těchto pulsů mezi přijatými rámci. Z dlouhodobého hlediska se tyto délky při zachování stejného „tick time“ mohou lišit až o $\pm 20\%$, ale u dvou po sobě jdoucích rámců je hranice pro rozdíl časů 1,5625%. Vždy se porovnává aktuální s předchozím přijatým synchronizačním pulsem, v případě většího rozdílu pulsů než je povolená mez se považují předchozí data za neplatná. Vzhledem k použité metodě vyhodnocení dochází k zahození dvou rámců v případě jednoho neplatného.

2.3.2.2 Spolehlivost kontrolního součtu

Vzhledem k použití pouze 4 bitového kontrolního součtu u rychlého kanálu je potřeba vzít v úvahu spolehlivost detekce chyb v přenášených rámcích.

Dle normy je detekovatelnost chyb následující [6]:

- všechny jednobitové chyby
- všechny v sudém počtu bitů (nejdoucí po sobě)
- všechny po sobě jdoucí chyby v počtu bitů ≤ 4
- 87,5% po sobě jdoucích chyb v počtu bitů $= 5$
- 93,75% po sobě jdoucích chyb v počtu bitů > 5

2.3.3 Struktura datového rámce - pomalý kanál

Typicky se využívá pro přenos doplňkových informací. Přenáší se jako část rychlých zpráv, kdy se v každém přenášeném rychlém rámci nachází nibble S&C. Tento nibble obsahuje dva bity sériového pomalého kanálu. Z této dvojice bitů se skládá pomalá zpráva o délce 16 nebo 18 bitů, dle implementace v konkrétní aplikaci.

Tab. 4: Nibble Status a Komunikace[6]

Číslo bitu	Význam bitu
0 (LSB)	dle aplikace
1	dle aplikace
2	bit pomalé zprávy
3 (MSB)	bit pomalé zprávy

2.3.3.1 Krátká pomalá zpráva

Tento formát pomalé zprávy se skládá z 16 rychlých zpráv.

Obsah zprávy:

- 4 bity ID
- 8 bitů dat
- 4 bity CRC

Tab. 5: Skladba krátké pomalé zprávy[6]

Pořadí přijatého nibblu (zprávy)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Start bit (bit 3 dle tab.4)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pomalá zpráva (bit 2 dle tab.4)	ID zprávy (3-0)				DATA (7-0)							CRC (3-0)				

2.3.3.2 Rozšířená pomalá zpráva

Tento formát pomalé zprávy se skládá z 18 rychlých zpráv. Jsou možné dvě varianty skladby obsahu zpráv.

Obsah zprávy - varianta 1 (C = 0):

- 8 bitů ID
- 12 bitů dat
- 6 bitů CRC
- 1 bit značící variantu

Obsah zprávy - varianta 2 (C = 1):

- 4 bity ID
- 16 bitů dat
- 6 bitů CRC
- 1 bit značící variantu

Tab. 6: Skladba rozšířené pomalé zprávy[6]

Pořadí přijatého nibblu (zprávy)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Pomalá zpráva (bit 3 dle tab.4)	1	1	1	1	1	1	0	C	8-bit ID (7-4) / 4-bit ID(3-0)				0	8-bit ID (3-0) / 4-bit DATA(15-12)				0
Pomalá zpráva (bit 2 dle tab.4)	6-bit CRC						12-bit DATA(11-0)											

V tabulce 7 jsou příklady identifikátorů, které jsou pevně dané normou. Mimo ty, které jsou zmíněné v tabulce norma definuje řadu dalších pro přenos doplňkových dat. Část rozsahu je definovatelná výrobcem. Jedním z identifikátorů je také typ senzoru. V případě tohoto identifikátoru je normou dána řada různých typů senzorů, kdy výrobce musí pro svou aplikaci zvolit patřičný identifikátor. Další z mnoha pevně definovaných částí protokolu je kód výrobce. Normou jsou pevně přiřazeny kódy jednotlivých výrobců například Continental, Bosch, Hella, Autoliv, Elmos, Freescale, NXP Semiconductor, ST Microelectronics, OnSemi, atd.

Tab. 7: Příklad ID pomalých zpráv (8-bit ID)[6]

ID Zprávy	Význam
0x01	Chybové a stavové kódy
0x03	Typ senzoru
0x05	Kód výrobce
0x06	Verze SENT protokolu
0x29	ID senzoru 1(LSB)
0x2A	ID senzoru 2
0x2B	ID senzoru 3
0x2C	ID senzoru 4 (MSB)

2.3.4 Multiplexování rychlého kanálu

V rámci poslední revize normy SAE J2716 APR2016, byla přidána možnost pro přenášení většího množství dat v rámci rychlého kanálu bez nutnosti zásahu do protokolu jako takového. Toho bylo docíleno nahrazením prvního datového nibblu za nibble „frame control - FC“.

Pro každou hodnotu v tomto řídicím nibblu je v aplikaci přiřazena patřičná sestava následujících až pěti nibblů. Tímto vzniknul prostor pro přenos až 80 nibblů dat z původních možných 6 nibblů. Samozřejmě tato modifikace je na úkor frekvence s jakou jsou zasílána jednotlivá data ze senzoru. V rámci multiplexování zpráv byla přidána další funkce a to „data consistency counter - DCC“, který slouží pro zajištění konzistence přijímaných dat. Norma dále definuje to jak je tento čítač inkrementován, k inkrementování hodnoty DCC dochází vždy když je zasílán rámec se stejnou hodnotou FC nibblu, který už při aktivní hodnotě DCC byl zaslán (příklad v tab.8).

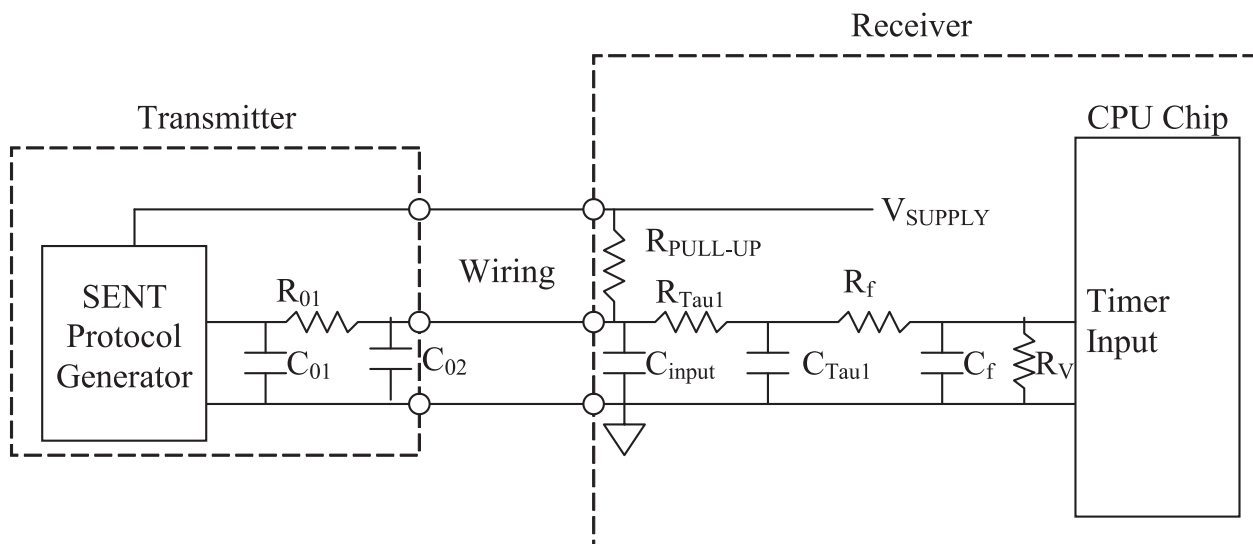
Tab. 8: Příklad pro teplotní senzor se 4 termočlánky, 16 bitové hodnoty

Zpráva	FC	DCC	Data 1 (MSN)	Data 2	Data 3	Data 4 (LSN)	
1	0	0	TC1 (15-12)	TC1 (11-8)	TC1 (7-4)	TC1 (3-0)	Data z termočlánků $t = n$
2	1	0	TC2 (15-12)	TC2 (11-8)	TC2 (7-4)	TC2 (3-0)	
3	2	0	TC3 (15-12)	TC3 (11-8)	TC3 (7-4)	TC3 (3-0)	
4	3	0	TC4 (15-12)	TC4 (11-8)	TC4 (7-4)	TC4 (3-0)	
5	0	1	TC1 (15-12)	TC1 (11-8)	TC1 (7-4)	TC1 (3-0)	Data z termočlánků $t = n+1$
6	1	1	TC2 (15-12)	TC2 (11-8)	TC2 (7-4)	TC2 (3-0)	
7	2	1	TC3 (15-12)	TC3 (11-8)	TC3 (7-4)	TC3 (3-0)	
8	3	1	TC4 (15-12)	TC4 (11-8)	TC4 (7-4)	TC4 (3-0)	

2.3.5 Fyzická vrstva

V úvodu bylo zmíněno že se jedná o komunikační rozhraní navržené s ohledem na nízké náklady, což s musí projevit v parametrech fyzické vrstvy.

Vzhledem k požadavku na nízké implementační náklady je možné použít například v MCU či ASIC RC oscilátor. Využití takového oscilátoru je mnohonásobně levnější než použití libovolného kvalitnějšího zdroje hodinového signálu, jako například krystal či specializovaný obvod pro generování hodinového signálu. Daní za minimální náklady je to, že takový oscilátor je velice nestabilní v čase a teplotách. Proto bylo nutné pro SENT definovat odchylku v „tick time“ $\pm 20\%$. Tato odchylka je ve smyslu dlouhodobého odchylení od výchozí hodnoty. Vzhledem k tomu se provádí synchronizace viz kapitola 2.3.2.1, odchylka mezi po sobě jdoucími rámci může být maximálně 1,5625% a odchylka v jednom rámci může být maximálně 0,37%.



Obr. 2: Příklad možné implementace rozhraní SENT[6]

2.4 CAN

„Controller Area Network - CAN“ je komunikační sběrnice využívaná v automobilovém průmyslu primárně pro komunikaci senzorů a aktuátorů s ECU, či mezi různými ECU. Obecně je možné CAN sběrnici vzhledem k jejím parametrům očekávat tam, kde je nutná komunikace mezi kritickými systémy.

Komunikace na sběrnici CAN je definována sadou norem ISO11898-1 až 6. Norma definuje sběrnici na úrovni fyzické a linkové vrstvy dle OSI modelu.

Základní technické parametry:

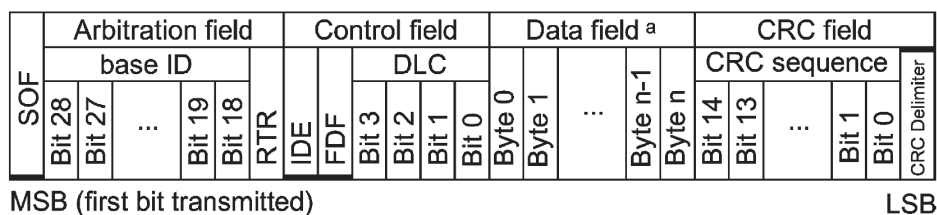
- čtyřvodičové připojení senzoru - Vcc, GND, CANH, CANL
- napěťové úrovně odvozené od 5V
- diferenciální přenos dat
- přenosová rychlost až 1Mbit/s (Classic CAN, CAN FD 5 Mbit/s[2])
- 0-64 bajtů dat v jednom datovém rámci (CAN FD)
- multi-master prioritní režim přístupu ke sběrnici [1]
- všechny přenosy jako „broadcast“ [1]
- vysoká odolnost

- detekce chyb a jejich signalizace [1]

2.4.1 Datové rámce

Norma ISO11898 specifikuje 5 možných datových rámců. Pro nové aplikace využívající CAN se doporučuje implementace Flexible Data rate (dále jen FD) rámců.

- Classic Base rámec - 11 bitů ID, 0 až 8 bajtů dat, 15 bitů CRC
- Classic Extended rámec - 29 bitů ID, 0 až 8 bajtů dat, 15 bitů CRC
- FD Base rámec - 11 bitů ID, 0 až 16 bajtů dat, 17 bitů CRC
- FD Base rámec - 11 bitů ID, 20 až 64 bajtů dat, 21 bitů CRC
- FD Extended rámec - 29 bitů ID, 0 až 16 bajtů dat, 17 bitů CRC
- FD Extended rámec - 29 bitů ID, 20 až 64 bajtů dat, 21 bitů CRC



Obr. 3: CAN Classic Base rámec [1]

2.4.2 Arbitráž na sběrnici

Při vysílání na CAN sběrnici může docházet ke kolizím. Fyzická vrstva sběrnice je navržena takovým způsobem, že i přes to že k těmto kolizím dochází nedojde k poškození přenášených dat. V případě že začne více zařízení vysílat ve stejných okamžik musí dojít při přenosu ID k rozdílu v přenášených bitech (alespoň jedno zařízení má výstupní stav „dominant“ a zbylá „recessive“) a v tu chvíli zařízení s nižší prioritou, tedy ta která aktuálně mají výstupní stav „recessive“, přeruší další přenos a vyčkají na uvolnění sběrnice. Vzhledem k tomu že žádná dvě zařízení nesmějí vysílat datové rámce se stejným ID je tímto způsobem zajištěno, že nedojde k poškození přenášených dat.

2.4.3 Acknowledgement - Potvrzování komunikace

Na konci každého přenášeného rámce je prostor nazývaný „ACK field“, který slouží pro potvrzení přijetí. V „ACK field“ příjemci vysílají potvrzení o přijetí, v případě přijetí poškozené zprávy je po „ACK field“ vysílán chybový rámec, který zajistí to že všichni příjemci na sběrnici přijatou zprávu zahodí a vysílač se pokouší o další přenos.

2.4.4 Bit stuffing na sběrnici

Metoda pro kódování rámců využívající metodu NRZ. Při přenosu pěti a více totožných bitů po sobě dochází k zařazení tzv. „stuff“ bitů. Díky tomu není možné aby se na sběrnici vyskytlo více než pět totožných bitů po sobě, jelikož „stuff“ bit je vždy opačné hodnoty než pět předchozích bitů. Tímto je dána maximální doba mezi jednotlivými hranami při překlápění logických úrovní. Tohoto se využívá pro re-synchronizaci všech jednotek na CAN sběrnici.

2.4.5 Detekce chyb

Sběrnice CAN byla navržena s ohledem na vysokou úroveň bezpečnosti komunikace ve smyslu detekování a eliminace chyb, nikoli z hlediska šifrování dat.

Sběrnice vykazuje vysokou odolnost díky vícenásobné detekci chyb.

Způsoby detekce chyb:

- monitoring sběrnice - vysílač kontroluje zda se vysílaný bit shoduje se skutečným bitem na sběrnici
- CRC - s každou zprávou se vysílá i CRC pro kontrolu správnosti přijatých dat
- ACK kontrola - každá zpráva je všemi příjemci potvrzována
- kontrola formátu - zpráva na sběrnici má přesně daný formát
- „bit stuffing“ - za každých 5 totožných bitů se vkládá opačný bit

2.5 LIN

„Local Interconnect Network - LIN“ je komunikační sběrnice využívaná v automobilovém průmyslu. Jedná se o sběrnici, která se využívá k obsluze senzorů a aktuátorů obdobně jako u sběrnice CAN, ovšem v případě sběrnice LIN se nejedná o obsluhu kritických systémů. U této komunikační sběrnice nebyl kladen důraz na úroveň spolehlivosti jako pro CAN, proto je implementace této komunikační sběrnice výrazně levnější než v případě sběrnice CAN.

Sběrnice LIN je definována sadou norem ISO17987-1 až 7.

Základní technické parametry:

- třívodičové připojení senzoru - Vcc, GND, LIN
- napěťové úrovně komunikace dle Vcc (typicky 12V a 24V)
- přenosová rychlost 1 až 20 kbit/s
- „master-slave“ komunikace

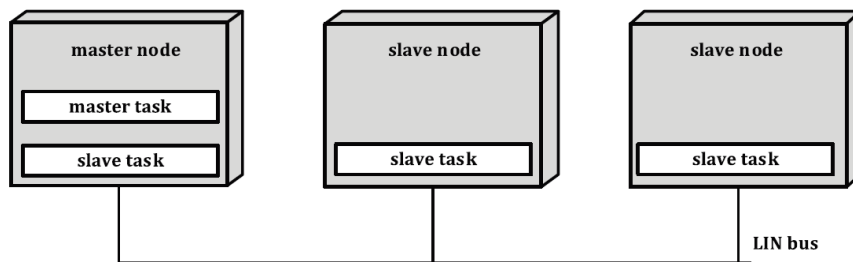
- až 16 „slave“ zařízení
- časové deterministická komunikace
- 1 až 8 bajtů dat v datovém rámci

2.5.1 Složení zařízení na LIN sběrnici

Sběrnice LIN se skládá vždy z jednoho master zařízení a až patnácti slave zařízení. Master zařízení zajišťuje tzv. „master task“ a „slave task“. Slave zařízení obsahují pouze „slave task“.

„Master task“ rozhoduje o dění na sběrnici, respektive o tom kdy a jaký datový rámec bude přenášén.

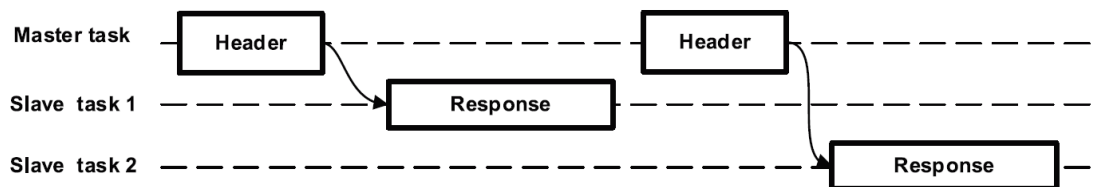
„Slave task“ poskytuje datové rámce na základě požadavku od „master tasku“.



Obr. 4: Rozložení master a slave úloh v zařízeních na sběrnici [3]

2.5.2 Komunikační rámce

Z principu master-slave komunikace je dáno že slave zařízení odpovídá na zprávy zasílané master zařízením. V případě sběrnice LIN komunikace probíhá tak, že master zařízení vyšle na sběrnici tzv. „header“ a slave zařízení na tento požadavek odpoví datovým rámcem.



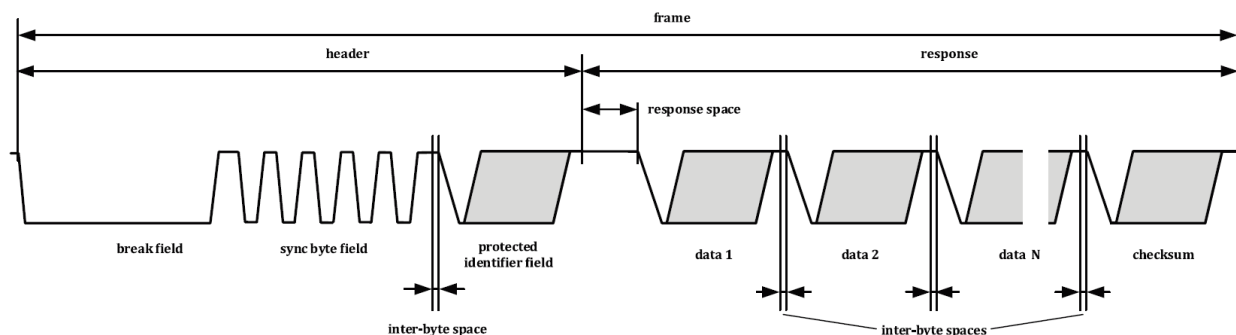
Obr. 5: Vysílání Headeru a odpovědí [3]

„Header“, který je vysílán master zařízením se skládá z tzv. „break field“, synchronizačního bajtu a chráněného ID. „Break field“ slouží k signalizaci počátku přenosu. Chráněné ID je bajt,

který obsahuje 6 bitů ID a 2 paritní bity.

Rozdělení ID dle normy [3]

- 0 až 59 - ID pro datové rámce
- 60 a 61 - přenos diagnostických a konfiguračních dat
- 62 a 63 - rezervováno pro budoucí rozšíření protokolu



Obr. 6: Header a datový rámec [3]

2.5.3 Časová determinističnost

Komunikace na sběrnici funguje časově deterministicky, jelikož komunikace je řízena master zařízením, které zahajuje komunikaci v pevně daných časových okamžicích definovaných časovou základnou (typicky 5 ms nebo 10 ms. Volba konkrétního vysílaného „headeru“ je definována plánovací tabulky v master zařízení, díky tomu nemůže dojít k přetížení sběrnice. Výjimkou jsou události řízené rámce, které jsou nadřazené nad plánovací tabulkou a jsou tedy vysílány v nejbližším možném signálu od hodinové základny.[3]

2.5.4 Kontrolní součet

Poslední byte každého datového rámce obsahuje kontrolní součet. A to buď v klasické nebo rozšířené verzi.

- standardní kontrolní součet - kontrolní součet z dat
- rozšířený kontrolní součet - kontrolní součet z dat a ID

2.6 PSI5

„Peripheral sensor interface 5 - PSI5“ je komunikační sběrnice využívaná v automobilovém průmyslu primárně pro periferní senzory airbagů. PSI5 je volně využitelná sběrnice, za kterou stojí PSI5 organizace.



Obr. 7: Členové PSI5 organizace [5]

Specifikace pro sběrnici PSI5 je definována základní obecnou specifikací a následnými rozšířeními pro oblasti „Airbag“, „Chassis and Safety“ a „Powertrain“.

U této sběrnice je kladen důraz na zajištění vysoké datové propustnosti, spolehlivosti a minimálních nákladů na implementaci.

Základní technické parametry:

- dvou vodičové připojení senzoru - V_{CC} , GND
- komunikace senzor -> ECU: proudově modulovaná data
- komunikace ECU -> senzor: napětově modulovaná data
- přenosová rychlost 125kbit/s, volitelně 189kbit/s
- kódování dat - Manchester kód
- synchronní a asynchronní přenos
- v případě synchronního přenosu master-slave komunikace
- vysoká odolnost a nízké vyzařování z hlediska EMC

2.6.1 Komunikace senzoru s ECU

Komunikace senzorů s ECU probíhá pomocí proudové modulace dat na napájecí linku senzorů, vzhledem k tomu je pro senzor nutnost použít pouze 2 vodiče pro připojení napájení 5V (5V napájení pevně dáno standardem) a GND. Toto řešení v sobě skrývá jisté výhody i nevýhody, mezi výhody lze řadit nutnost pouze dvou vodičů. Mezi nevýhody lze zařadit potřebu 5V napájení, které musí ECU poskytnout pro senzory.

Pro samotný přenos dat se využívá manchester kódování. Při využití tohoto principu kódování dat není hodnota bitu dána přímo logickou úrovní jako takovou, ale náběžnou či sestupnou hranou v polovině času bitu. Je možné využít dvou různých možností interpretace dat. Původní interpretace podle autora Manchester kódu Dr. Thomase kdy náběžná hrana v polovině bitu reprezentuje logickou 0 a sestupná hrana logickou 1. Pozdější interpretace dle IEEE 802.3 je opačná tedy náběžná hrana reprezentuje logickou 1 a sestupná hrana logickou 0.

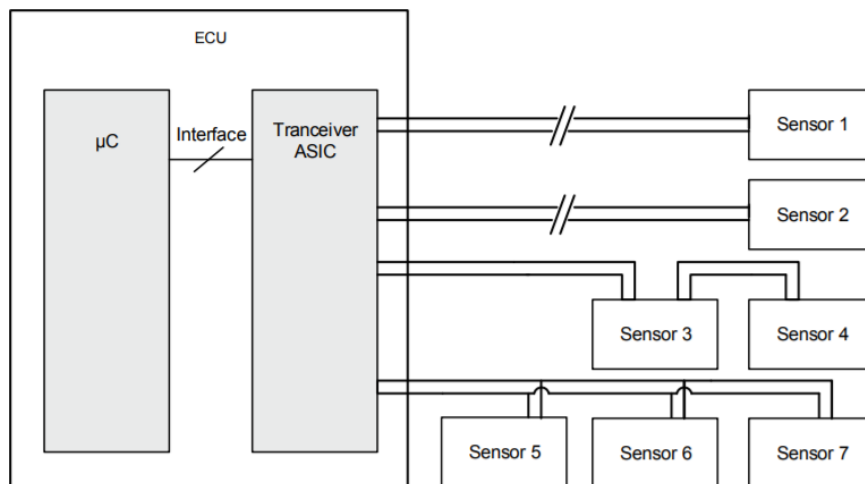
2.6.2 Komunikace ECU se senzorem

Komunikace ECU se senzory je možná pomocí napěťových pulsů na napájecí lince senzorů. Standardně se těchto pulsů využívá pro synchronizaci, ale je možné ji využít i pro komunikaci se senzory. Pro přenos dat je možné použít dva způsoby na úrovni fyzické vrstvy. „Tooth-gap“ a „Pulse-width“ metody. V případě „tooth-gap“ je logická 0 definována krátkým synchronizačním pulsem a mezerou a logická 1 je dána dvěma krátkými synchronizačními pulsy. V případě „pulse-width“ je logická 0 definována krátkým synchronizačním pulsem a logická 1 dlouhým synchronizačním pulsem.

2.6.3 Topologie propojení senzorů s ECU

PSI5 se může chovat jako sběrnice i jako rozhraní. Konfigurací senzoru je možné standardně volit synchronní či asynchronní režim komunikace čímž se definuje zda je možné PSI5 využít jako sběrnici či pouze jako rozhraní. Při asynchronním režimu je možné PSI5 využívat pouze jako rozhraní pro jeden senzor. V případě využití v synchronním režimu je taktéž možné PSI5 využít jako rozhraní, ale výrazně výhodnější může být použití jako sběrnice.

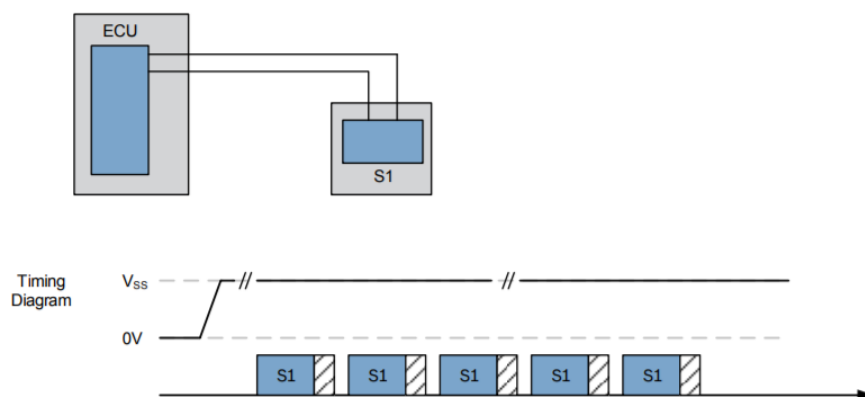
- paralelní - senzory jsou paralelně propojeny uvnitř ECU
- „pass-through“ - senzory jsou paralelně propojeny uvnitř senzoru
- „splice“ - paralelní propojení senzorů se provádí odbočkami na kabeláži
- „Daisy-Chain“ - zřetězení senzorů, kdy každý senzor spíná dále připojené senzory



Obr. 8: Možná propojení ECU a senzorů [4]

2.6.4 Asynchronní mód

Asynchronní režim podporuje pouze jediný možný způsob propojení senzoru s ECU vzhledem k tomu, že se v tomto režimu jedná pouze o interface. Komunikace v tomto režimu probíhá tak, že senzor periodicky zasílá data do řídicí jednotky, periodičita zasílání je řízena senzorem.

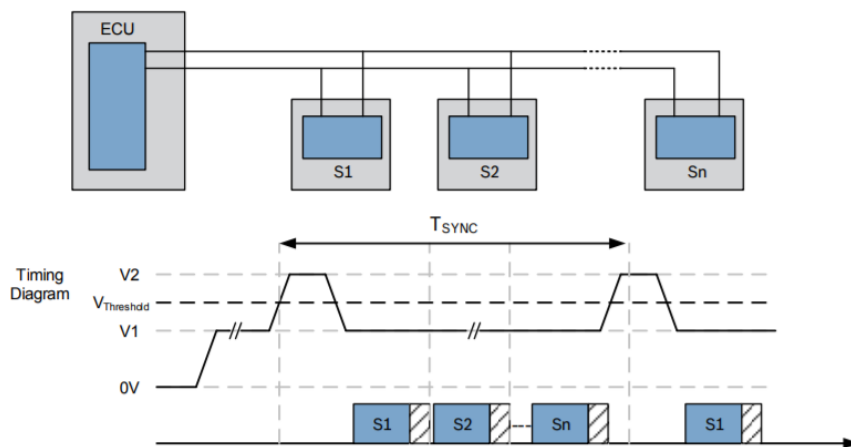


Obr. 9: Propojení ECU se senzorem v asynchroním režimu a časový diagram komunikace [4]

2.6.5 Synchronní (sběrníkový) mód

Synchronní režim je rozsáhlejší. Propojení senzorů s ECU je možné realizovat více způsoby. Komunikace je v tomto režimu řízena synchronizačními pulsy vysílanými z ECU, kdy po každém synchronizačním pulsu senzory zasílají data do ECU. Synchronizačním pulsem se rozumí napěťový skok na napájecí lince senzorů. Zasílání dat ze senzorů probíhá vždy v přesně daném časovém okně

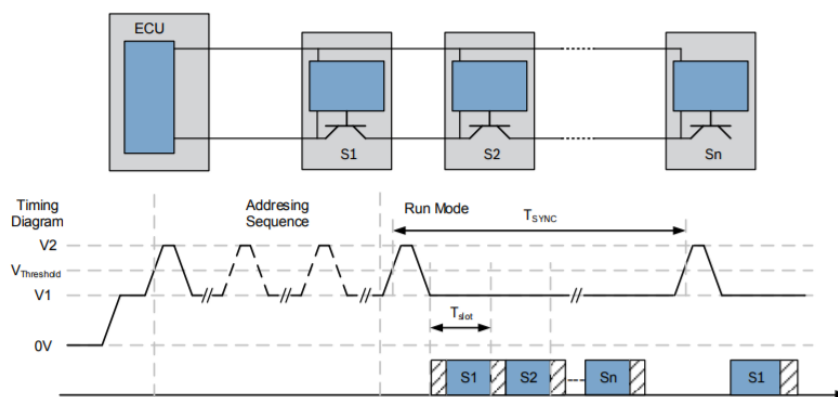
po synchronizačním pulsu, které je pro každý senzor přiděleno, respektive pro které časové okno je senzor nakonfigurován. Tímto způsobem je možné připojit až 6 senzorů.



Obr. 10: Propojení ECU se senzory v synchroním režimu a časový diagram komunikace [4]

2.6.6 Daisy-chain mód

Daisy-chain mód má obdobné možnosti a vlastnosti jako standardní synchronní režim komunikace, ale v tomto případě je jiný způsob rozdělení časových oken v komunikačním rámci. Oproti synchronnímu režimu kdy má každý senzor pevně přidělené časové okno se u daisy-chain propojení po každém startu ECU přidělují časové okna dle aktuálního propojení senzorů, tedy je možné senzory libovolně zaměnit.



Obr. 11: Propojení ECU se senzory v daisy-chain režimu a časový diagram komunikace [4]

2.7 Analogová rozhraní

Jednou z variant pro přenos dat je možné využít analogových rozhraní. Jedná se o řešení, která jsou z hlediska inovací a nároků z velké většiny již zastaralá, ovšem i přes to hrála zásadní roli v senzorických systémech. V dnešní době se používají již výhradně digitální sběrnice a systémy.

Napěťový výstup: Jednou z nejtýpčtějších variant analogových rozhraní je využití napěťového výstupu, kdy je přenášená hodnota reprezentována napětovou úrovní. Napěťová úroveň je dle předem definované převodní charakteristiky převedena na patřičnou hodnotu. Jedná se o poměrně jednoduchý princip, který se ovšem v praktických aplikacích setkává se značnými problémy ve formě problematické odolnosti vůči rušení. Odolnost takto přenášeného signálu je závislá na více faktorech. Patří mezi ně například kvalita kabelu pro přenos signálu, počet kvantovacích úrovní vůči napěťovému rozsahu výstupního signálu a zatížitelnost zdroje signálu.

PWM výstup: Druhým zástupcem pro analogová rozhraní je signál modulovaný pulsně šířkovou modulací. V tomto případě je signál reprezentován střídou PWM signálu. Vzhledem k využití pouze dvou napěťových úrovní je zaručena vyšší odolnost, ovšem díky využití obdélníkového signálu je takové rozhraní horší z hlediska vyzařování do okolí (závislé na ostroti náběžných a sestupných hran signálu). Výstupní hodnota je dána střídou signálu v rozsahu 0-100%, rozlišení které je pro daný signál je závislé na ostroti náběžných a sestupných hran signálu, kdy se zvyšující se ostroť hran je možné využití vyššího rozlišení, ovšem na úkor spektra vyzařovaného signálu.

2.8 Srovnání vybraných sběrnic a rozhraní

V tabulce 9 je přehled některých základních parametrů z popisovaných rozhraní a sběrnic využívaných v automobilovém průmyslu.

Tab. 9: Porovnání vybraných sběrnic a rozhraní

Sběrnice / Rozhraní	Datová propustnost	Maximální počet zařízení	Počet vodičů	Napěťové (Proudové) úrovně dat
SENT	až 24,7 - 64,9 kbit/s	2	3	5 V
CAN	1 Mbit/s Classic CAN (5 Mbit/s CAN FD)	typicky do 32 (při nižších rychlostech více)	4	5 V
LIN	20kbit/s	17	3	typicky dle V_{CC}
PSI5	typicky 125 kbit/s (189 kbit/s)	4	2	signál 4 - 19 mA (standard)

3 Poruchové stavy

3.1 EMC a možné vlivy na poruchové stavy

3.1.1 Obecné informace

Automobilová elektronika podstupuje náročné testy na EMC. Jedná se jak o testy na vyzařování, tak o testy na imunitu. Z hlediska vyzařování se jedná o požadavky na dodržení limitů pro vyzařované rušení na vedení a vzduchem. Z hlediska imunity jde o odolnost elektroniky vůči stavům, ke kterým v palubní síti a prostředí automobilů dochází.

Následuje výběr testů z oblasti EMC, které typicky mohou ovlivnit či způsobit některé z dále popisovaných chyb.

3.1.2 Přechodové jevy

Z hlediska odolnosti se jedná o jednu z nejnáročnějších oblastí testů. V palubní síti automobilu se mohou vyskytovat velmi vysoké výkyvy napětí řádově ve stovkách voltů. Testování na tyto přechodové jevy se provádí pomocí testovacích pulsů, které jsou přiváděny na napájecí (ISO7637-2 a ISO17650-2) a signálové linky (ISO7637-3) testované elektroniky.

Tyto přechodové jevy mají samozřejmě vliv na elektroniku automobilu. Pro každý z testovacích pulsů jsou stanoveny třídy, ty definují jak se musí testovaná elektronika během a po skončení testovacího pulsu chovat. U některých zařízení je požadována plná funkčnost, u jiných méně kritických je možnost výpadků, či nepřesností v měření, atd.

3.1.3 Rušivé signály na vedení

„Line Impedance Stabilization Network - LISN“ je tzv. umělá síť. Jedná se o zařízení, které se využívá pro testování na přenos rušení po napájecím vedení. Limity rušení jsou dány pro automobilové aplikace normou CISPR 25 ve frekvenčním pásmu 150 kHz až 108 MHz (napěťová metoda, při použití proudové sondy se provádí měření až do 245 MHz).

V naprosté většině elektronických systémů se využívá digitálních obvodů, což se typicky vyznačuje přepínáním logických úrovní na vysokých frekvencích. Často se jedná o aplikace využívající mikrokontrolér a krystal. Dalším zdrojem rušení bývají spínané zdroje, jejichž spínací frekvence se typicky pohybuje ve spodní části měřeného frekvenčního pásma (značná část zdrojů využívá spínací frekvenci v oblasti 200 - 500 kHz). V neposlední řadě se dnes výhradně využívají digitální komunikační rozhraní či sběrnice, které se taktéž vyznačují značnou strmostí náběžných a sestupných hran. Toto všechno a různé další vlivy se projevují jako rušení na napájecím vedení (často se ve spektru rušivého signálu vyskytují násobky hodinového signálu mikrokontroléru - jako 64MHz, 96MHz a další).

V případě nedodržení stanovených limitů pro rušení, by mohlo docházet k negativnímu ovlivnění některého z dalších zařízení.

3.2 Obecné informace o poruchových stavech

Pro automobilový průmysl je definována řada norem a nařízení, které rozřazují elektronická zařízení automobilu do různých tříd. Rozdělení do tříd je provedeno podle požadavků na bezpečnost, tedy čím kritičtější systém tím vyšší nároky jsou na něj kladeny. Tyto nároky jsou definovány funkčními třídami pro testovanou elektroniku.

Čím vyšší funkční třída je vyžadována tím se zmenšuje prostor pro situace, při kterých by mohlo dojít k chybě v elektronickém systému. Z hlediska vysokoteplotních senzorů, na jejichž simulaci je tato práce zaměřena, se obvykle nejedná o vysoce kritické systémy, které nemají přímý podíl na bezpečnosti posádky automobilu, proto jsou limity výrazně mírnější než například u bezpečnostních prvků typu airbag, brzdy a podobně. Nicméně to neznamená, že by na vysokoteplotní senzory nebyly kladeny vysoké nároky na spolehlivost.

Na toto jsou navázány poruchové stavy, které jsou definovány jako stavy kdy se zařízení dostává mimo standardní provozní stav. Výskyt poruchových stavů je mimo jiné zapříčiněn vlivy různých přechodných stavů. Základní přehled přechodných jevů, které mohou mít vliv na vznik poruchových stavů, je popsán v kapitole 3.1.

3.3 Rozdělení poruchových stavů senzorů v automobilovém průmyslu

3.3.1 Stavová hlášení senzoru

V případě detekování stavu, který je mimo požadované provozní podmínky, či došlo k nějaké detekovatelné poruše senzoru, vysílá senzor tzv. FMI. Mezi FMI kódy se zařazují informace o přepětí nebo podpětí v palubní síti. Další parametr, který je značně ovlivněn okolním prostředím je hlášení o překročení horního či spodního teplotního limitu pro elektroniku senzoru, kdy se typicky senzory provozují v teplotách od -40°C až do $+150^{\circ}\text{C}$. Mezi FMI kódy se řadí také stavové informace, respektive informace o poruše senzoru, nebo jeho části. Například z hlediska teplotních senzorů se může jednat o překročení horního či spodního limitu teploty na termočlátku (typicky od -40°C až do teplot okolo 1000°C). Další zasílané FMI kódy mohou odpovídat diagnostikovaným chybám jako je přerušování termočlátku či jeho zkrat, atd.

3.3.2 Chyby v datech

Za chyby v datech se považuje odchylka měření mimo toleranční pásmo kolem skutečné hodnoty. To se může projevovat dvěma způsoby. Prvním z nich jsou krátké extrémy (výrazně vyšší, či nižší

hodnota vůči předchozím a následujícím hodnotám) v měřené, respektive odesílané hodnotě. Druhým způsobem je dlouhodobá či trvalá odchylka.

Všechny chyby tohoto typu se vyznačují tím, že rámce jsou naprosto korektně přeneseny a k chybě či poruše dochází ještě před komunikačním rozhraní či sběrnici.

3.3.3 Chyby komunikačního protokolu

Za chyby komunikačního protokolu jsou považovány všechny chyby které způsobí nějaké poškození přenášené zprávy. Jedná se o kategorii chyb, do které spadá značná část všech chyb. Jedná se například o chybné bity, špatné CRC, změny v čase mezi zasíláním zpráv, a další.

3.3.4 Chyby fyzické vrstvy

Na úrovni fyzické vrstvy mohou vznikat dva typy chyb. Prvním typem je nesprávnost úrovní (typicky napěťových). Druhým typem je chyba v časování, kdy neodpovídají délky jednotlivých bitů, nebo například nejsou dodrženy limity pro strmost náběžných a sestupných hran.

4 Návrh a realizace zařízení

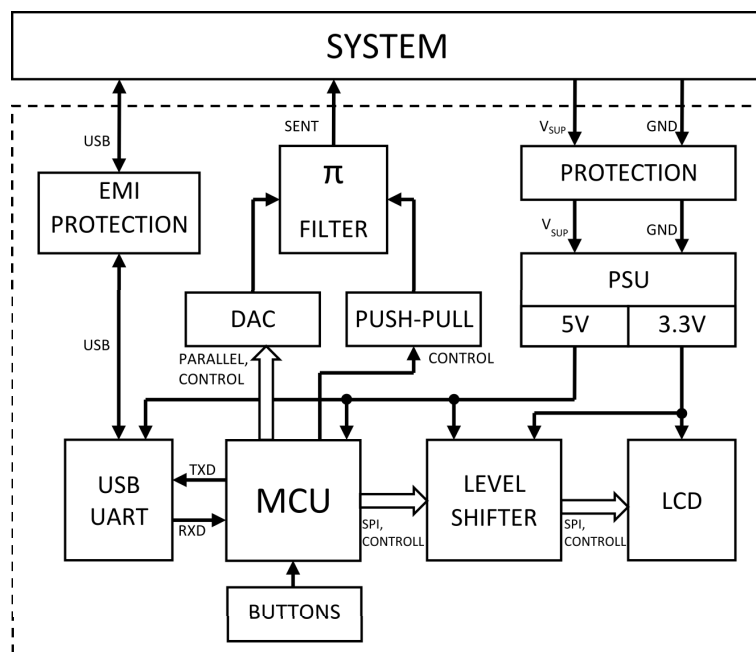
4.1 Požadavky na zařízení

Z hlediska praktické využitelnosti simulátoru vznikly následující požadavky.

- možnost generování libovolného typu senzoru využívané ve firmě Continental Powertrain Czech Republic (typy senzorů, respektive jejich komunikační protokoly, definovány normou SAE J2716-2016[6])
- věrná simulace funkcionality senzorů
- generování chybových stavů senzorů (FMI kódy, chyby v datech, chyby v protokolu)
- simulace opakování chyb dle SAE J2716-2016 příloha C.1.1[6]

4.2 Návrh HW

Návrh designu celého zařízení je rozčleněn na jednotlivé ucelené bloky, které zabezpečují určitou funkcionalitu.



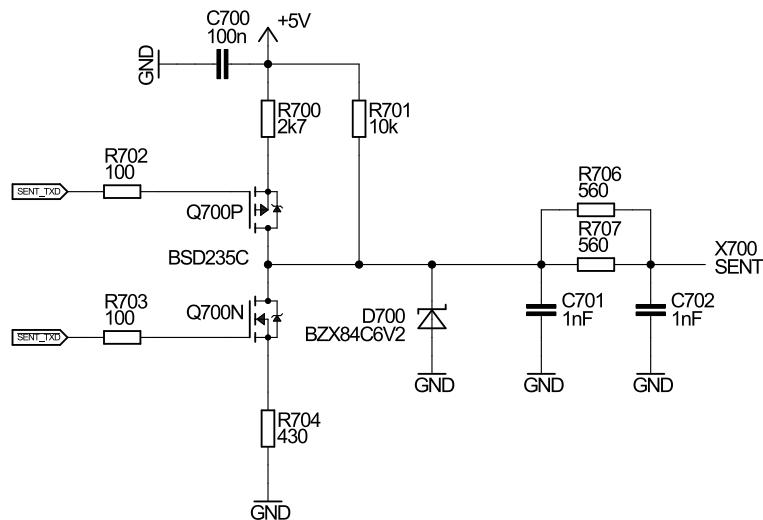
Obr. 12: Blokový diagram navrženého zařízení

4.2.1 SENT výstup

Pro zajištění funkcionality rozhraní SENT byly zvoleny dva možné přístupy. Společnou částí obou zapojení je samotný výstupní obvod. Topologie výstupního obvodu je doporučena v normě SAE J2716 (obrázek 2). Jedná se o pasivní filtr, konkrétně o π -článek skládající se ze dvou kondenzátorů a jednoho rezistoru. Využití tohoto výstupního filtru je nutné vzhledem k požadavkům z hlediska EMC u elektroniky pro automobilový průmysl. Tento výstupní filtr tvaruje signál tak, aby se omezila strmota náběžných a sestupných hran, díky tomu se sníží jak vyzařování RE, tak i úrovně vyšších harmonických na signálové lince.

4.2.1.1 Push-Pull výstup

První z možných topologií pro generování výstupního signálu je využití tzv. push-pull výstupního obvodu. Toto řešení se z hlediska návrhu senzoru jeví, jako velice dobré, jelikož je jednoduché a ponechává relativně minimální prostor pro potenciální problémy. Řízení tohoto výstupního obvodu je z hlediska mikrokontroléru snadné, jelikož se přivádí přímo obdélníkový signál z GPIO pinů na gate P-MOS a N-MOS tranzistorů.



Obr. 13: Návrh push-pull výstupního obvodu pro SENT

Za push-pull výstup se zařazuje tvarovací π -článek. Tento tvarovací článek slouží k zaoblení náběžných a sestupných hran, to je nutné z hlediska omezení vyzařování. Při příliš strmých náběžných a sestupných hranách by docházelo k vyzařování vyšších harmonických ze kterých se skládá obdélníkový výstupní signál. Při návrhu tohoto tvarovacího článku je nutné brát v úvahu parametry pro náběžné a sestupné hrany, napětové úrovně a další parametry definované normou.

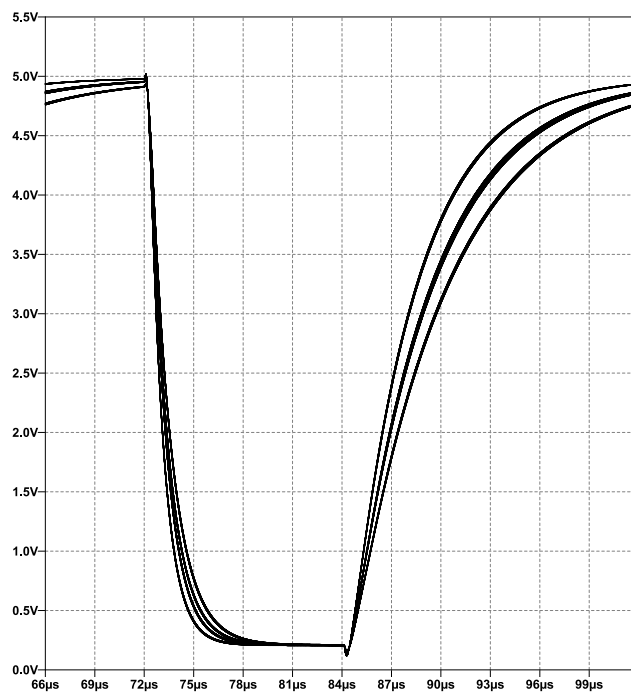
Tab. 10: Parametry definované normou[6]

Parametr	Minimum	Maximum	Jednotky	Podmínky
U_{OL}		0.5	V	
U_{OH}	4.1	5.5	V	
T_{FALL}		6.5	μs	$U_O = 3,8 \rightarrow 1,1 \text{ V}$
T_{RISE}		18	μs	$U_O = 1,1 \rightarrow 3,8 \text{ V}$
T_{STABLE}	6		μs	$U_O > 3,8 \text{ V} ; U_O < 1,39 \text{ V}$

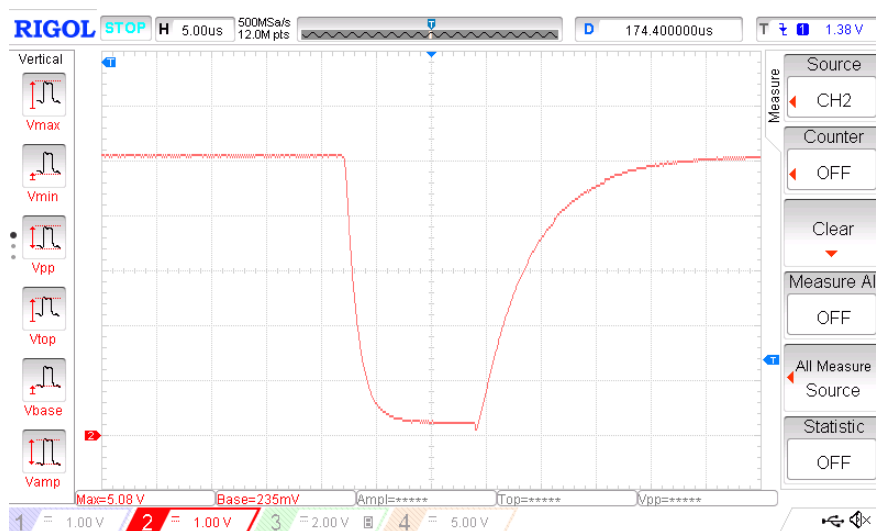
Simulace výstupního π -článku byla provedena s požadavkem na nalezení „worst case“. V rámci toleranční analýzy byly využity reálné hodnoty tolerancí pro rezistory 1% a pro keramické kondenzátory 20%. Do tolerancí nebylo možné zahrnout tolerance pro unipolární tranzistory v push-pull obvodu. V rámci tolerancí rezistorů a kondenzátorů vyšly následující parametry.

Tab. 11: Výsledky simulace a reálného měření

	Parametr	Minimum	Maximum	Jednotky	Podmínky
Simulace	Fall time	1,195	1,771	μs	$U_O = 3,8 \rightarrow 1,1 \text{ V}$
	Rise time	4,592	6,780	μs	$U_O = 1,1 \rightarrow 3,8 \text{ V}$
Měření	Fall time	1,48		μs	$U_O = 3,8 \rightarrow 1,1 \text{ V}$
	Rise time	5,64		μs	$U_O = 1,1 \rightarrow 3,8 \text{ V}$



Obr. 14: Simulace tvarovacího π -článku



Obr. 15: Reálný měřený průběh

Dle výsledků v tabulce 11 je zřejmé že parametry reálného obvodu odpovídají simulacím. Měřené hodnoty se pohybují přibližně ve středu mezi minimální a maximální hodnotou pro náběžné a sestupné hrany. Z toho lze usuzovat, že tolerance komponent se nepohybují v žádném z extrémů. Z hlediska požadavku na simulaci senzorů s různou hodnotou „tick time“ pro komunikaci na roz-

hraní SENT, bylo nutné navrhnout tvarovací článek tak, aby splnil požadavky na nejkratší hodnotu tohoto času. Z toho vyplývá, že pro vyšší hodnoty „tick time“ budou výrazně strmější hrany než by bylo vhodné, ovšem to je nutnost pro splnění nejkratších časů.

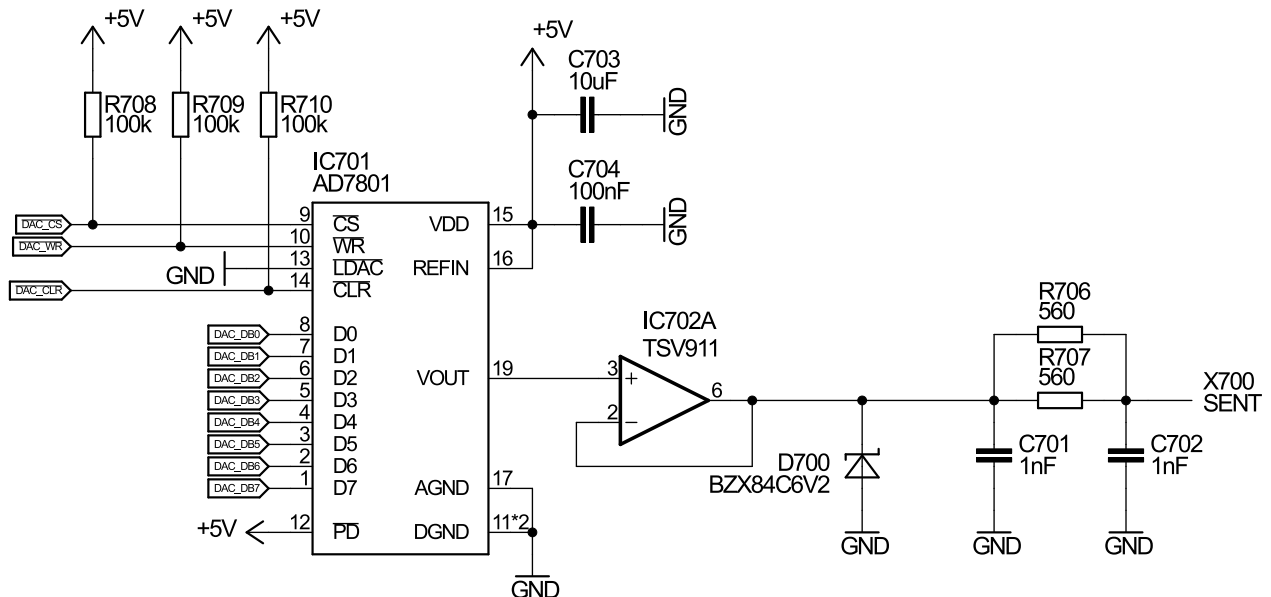
V případě požadavků na splnění limitů pro rušení na signálovém vedení by bylo nutné modifikovat HW strukturu výstupního tvarovače a to například připojováním dodatečných kondenzátorů k základním použitým.

4.2.1.2 DAC + Buffer

Druhou variantou pro generování výstupního signálu je využití rychlého DAC. Takové řešení by umožnilo měnit pomocí SW výstupní napěťové úrovně, náběžné a sestupné hrany. Daní za takovou funkcionalitu by byla značná komplikace řízení výstupního obvodu vůči první variantě.

Toto řešení se skládá z paralelního DAC a bufferu z operačního zesilovače s vysokým výstupním proudem ve funkci sledovače. Dále již následuje tvarovací π -článek obdobně jako u první varianty. U tohoto zapojení by bylo možné část tvarovacího obvodu vynechat jelikož tvarování signálu je částečně možné pomocí DAC, ovšem mimo to že výstupní π -článek tvaruje, tak plní také ochrannou funkci proti ESD.

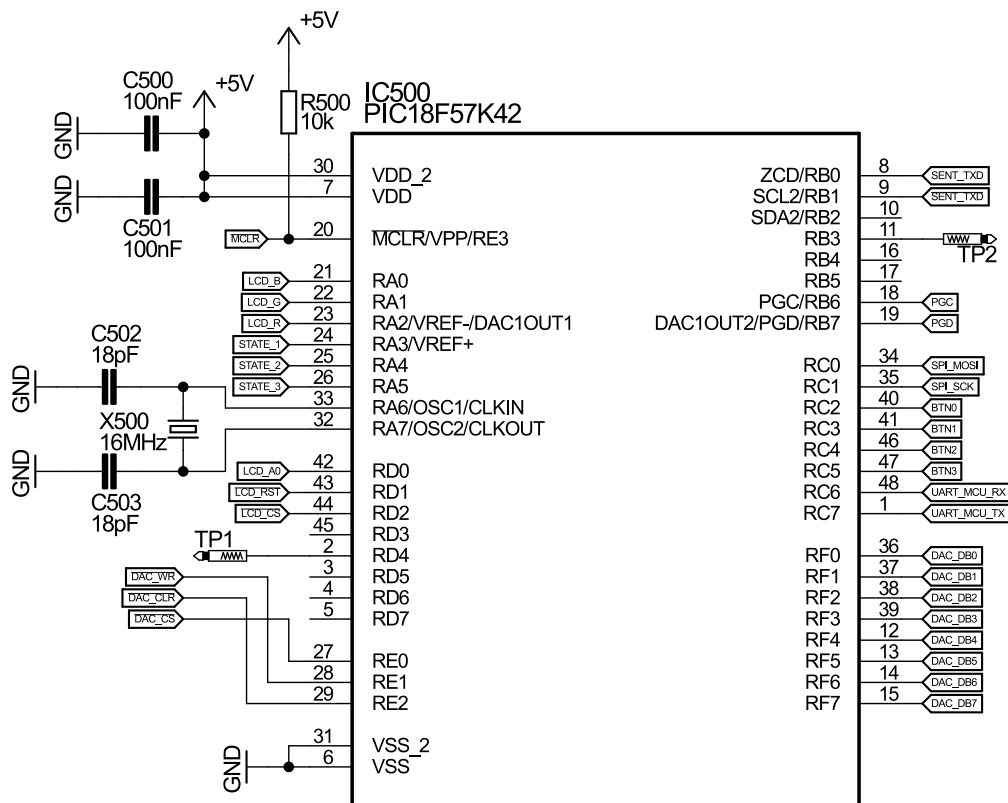
Nevýhodou tohoto řešení je navýšení náročnosti na výpočetní výkon, kdy by bylo nutné namísto přepnutí dvojice GPIO pinů bylo nutné přenést několik bajtů dat pro náběžnou či sestupnou hranu.



Obr. 16: Návrh DAC výstupního obvodu pro SENT

4.2.2 Mikrokontrolér

Pro řízení celého zařízení je zvolen mikrokontrolér firmy Microchip PIC18F57K42. Jedná se o mikrokontrolér z jedné z nejnovějších rodin v oblasti 8-bitových mikrokontrolérů. Mezi přednosti tohoto mikrokontroléru se řadí re-mapovatelnost většiny vstupů a výstupů periferních obvodů, mimo ty které jsou pevně zaznačeny ve schematické značce na obrázku 17. Další podstatnou výhodou je vůči předchozím řadám 8-bitových mikrokontrolérů navýšení kapacit paměti pro program i data, modernizace většiny starších periférií a dodání nových periférií.

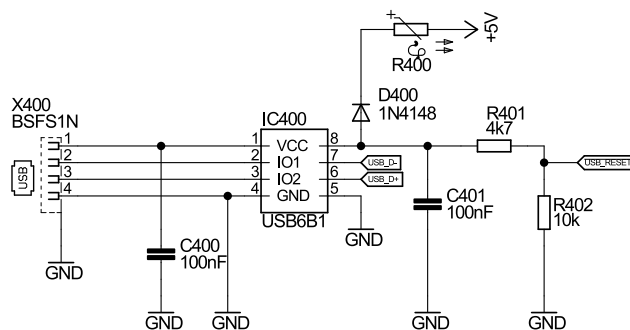


Obr. 17: Mikrokontrolér s okolními komponenty pro zajištění funkčnosti

4.2.3 USB

Připojení zařízení do počítače je provedeno pomocí obvodu FT232, který zajišťuje převod z USB na UART. V počítači zařízení figuruje jako virtuální COM port, se kterým se pracuje jako se standardním fyzickým RS232 portem.

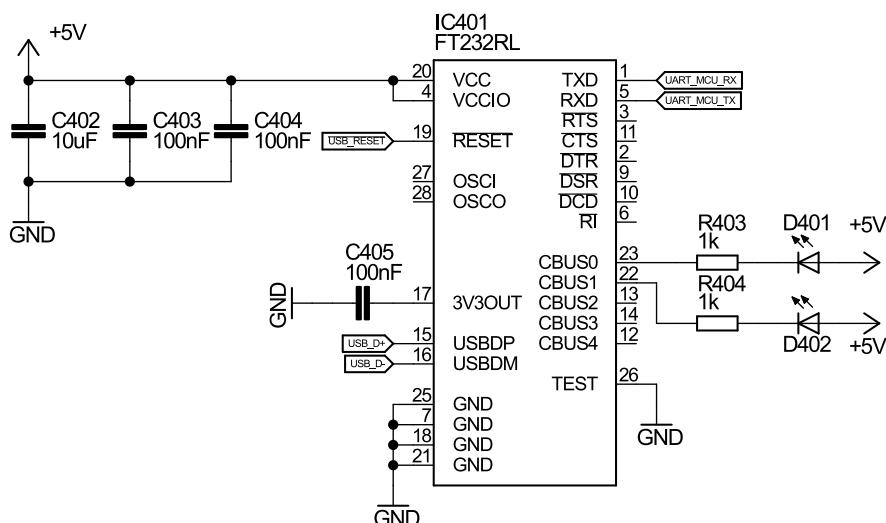
Ochrana proti ESD je zajištěna obvodem USB6B1, který integruje ochranu datových linek proti přepětí a TVS diodu pro ochranu napájecí linky.



Obr. 18: USB - vstupní ochrana proti ESD

V zařízení je využito standardního katalogového zapojení. Obvod FT232 je z resetu uveden do chodu po propojení USB portu simulátoru s PC.

V rámci této práce byla možnost využití mikrokontroléru s integrovanými periferními obvody pro připojení k USB, ale v rámci firemních standardů se využívá řešení s použitím převodníku FT232.



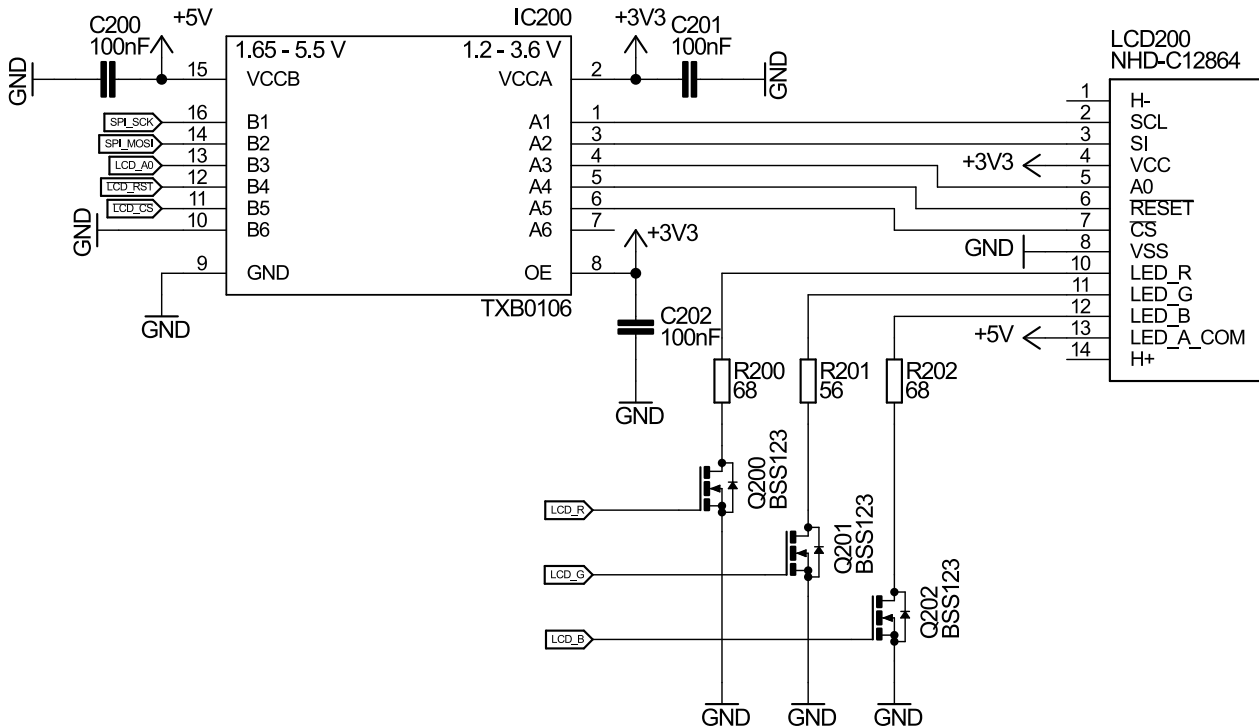
Obr. 19: USB - UART převodník

4.2.4 Ovládací a zobrazovací prvky

Pro zprostředkování informací uživateli je využit grafický displej s rozlišením 128x64 pixelů. Jedná se o grafický displej s grafickým řadičem ST7565. Grafický řadič pracuje s 3,3 V logikou. Jelikož je zbylá část zařízení napájena 5 V, pracuje na úrovni 5 V logiky. Proto je nutné zajistit napěťové přizpůsobení SPI pro komunikaci mikrokontroléru s displejem.

Pro napěťové přizpůsobení je využito obousměrného převodníku úrovní TXB0106, který dokáže převádět logické úrovně z napěťové úrovně z 1,65 - 5,5 V na 1,2 - 3,6 V dle připojených napěťových

úrovní na pinech V_{VCCA} a V_{VCCB} .

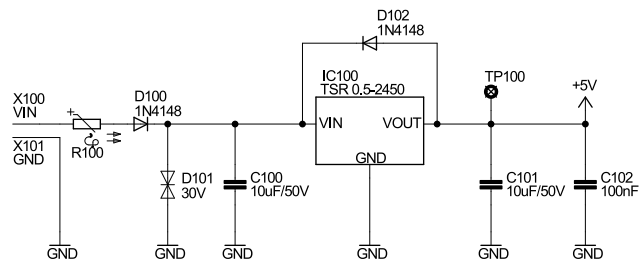


Obr. 20: Displej s převodníkem logických úrovní z 5 V na 3,3 V

4.2.5 Napájecí zdroj

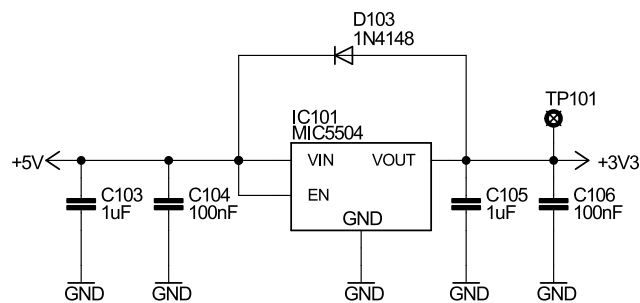
Pro napájení zařízení je potřeba zajistit hlavní napájecí větev 5 V. Z důvodu požadavku na široký rozsah napájecího napětí je využít spínaný regulátor.

Jako ochrana proti přepólování je využita usměrňovací dioda, jelikož se jedná o zařízení s velmi nízkým proudovým odběrem. V případě že by bylo potřeba zajistit napájení proudově náročnějšího zařízení bylo by vhodnější využít P-MOSFET z důvodu výrazně nižšího napětového úbytku na tranzistoru vůči standardní diodě (1N4148 má úbytek napětí v propustném směru až 1,6 V dle proudového zatížení, v případě P-MOSFETu je možné uvažovat řádově nižší úbytek vzhledem k velmi nízkému $R_{DS(on)}$). Další ochranu zajišťuje PTC termistor ve funkci vratné pojistky a TVS dioda pro ochranu proti přepětí. V této kombinaci zajišťuje ochranu zařízení proti zkratovému proudu a chrání TVS diodu proti zničení tepelnými ztrátami p potlačování napětových špiček.



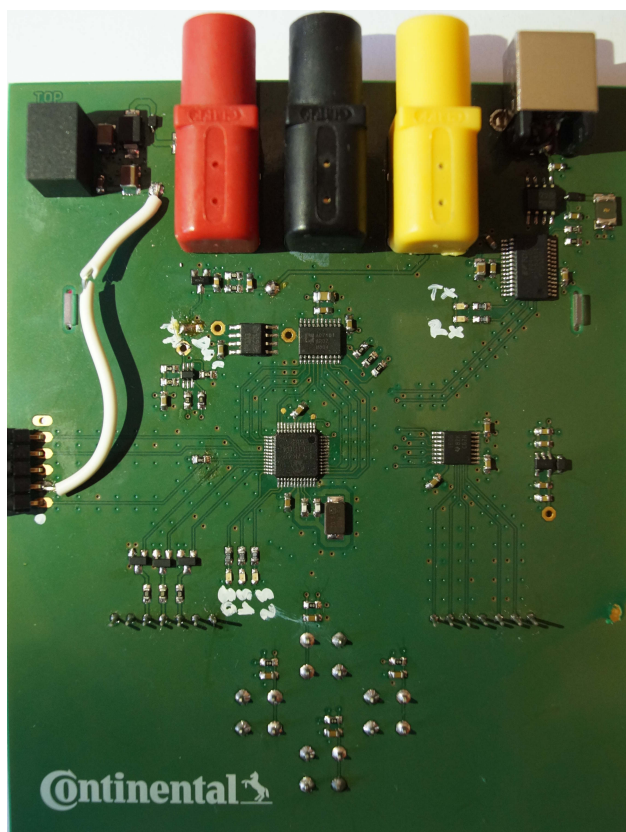
Obr. 21: Měnič z 7 - 28 V na 5 V pro napájení celého zařízení

Pro logiku grafického řadiče displeje a převodník úrovní je potřeba napájení 3,3V. Tato napájecí větev je odvozena od hlavní 5V větve. K tomuto účelu je využito LDO lineárního stabilizátoru.

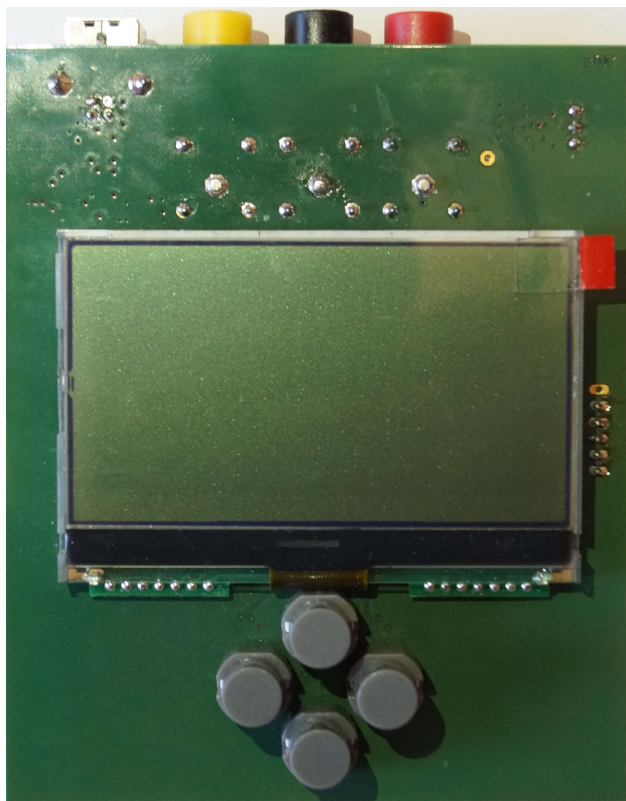


Obr. 22: Lineární regulátor z 5V na 3,3V pro logiku LCD

4.2.6 Realizovaný simulátor



Obr. 23: Fotka realizovaného simulátoru - strana TOP



Obr. 24: Fotka realizovaného simulátoru - strana BOT

4.3 Návrh FW

4.3.1 Návrh architektury FW

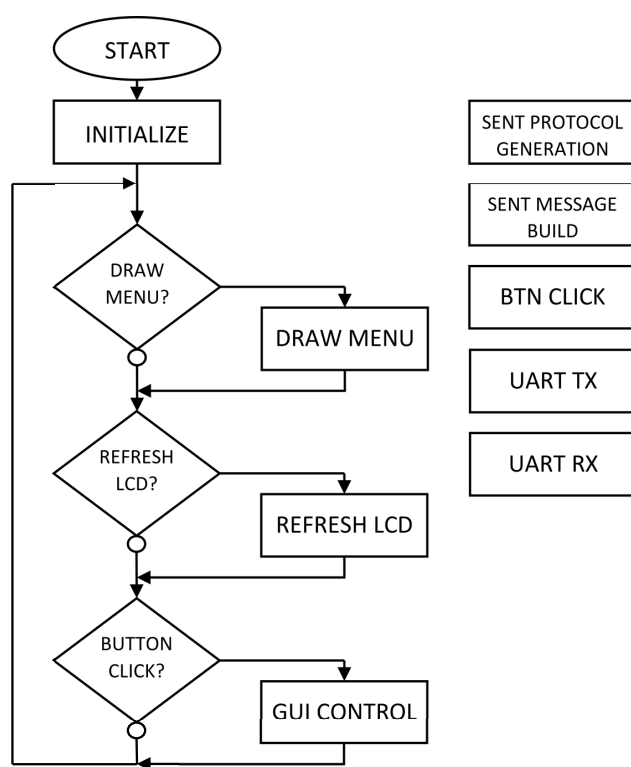
Vzhledem k navrhovanému systému, respektive k požadavkům kladeným na simulátor a nárokům na přesnost načasování jednotlivých systémových komponent bylo zvoleno pro celkový návrh architektury následující rozložení.

V rámci systému jsou tři úrovně, ve kterých se vykonává program. Nejnižší úroveň je hlavní smyčka, ve které se vykonávají nejméně kritické systémové komponenty, u kterých není konkrétní požadavek na odezvu. Druhou a třetí úroveň představuje zpracování obslužných rutin přerušení, to bylo dáno využitím mikrokontroléru s dvojicí úrovní priorit pro systém přerušení. Druhou úroveň pro zpracování programu tedy představují obslužné rutiny přerušení od zdrojů s nízkou prioritou. A třetí úroveň představují obslužné rutiny přerušení od zdrojů s vysokou prioritou.

Vzhledem k nutnosti přesného generování SENT protokolu bylo využito pro generování signálů přerušení s vysokou prioritou. Díky tomu je zajištěna minimální doba odezvy, jelikož dochází ke skokům na tuto obslužnou rutinu ať je vykonávána jakákoliv jiná část programu (skoky z hlavní smyčky a obslužných rutin přerušení od zdrojů s nízkou prioritou).

Na druhé úrovni probíhá vykonávání značně většího množství systémových komponent. Dochází k vyhodnocování stisků tlačítek, generování dat pro SENT protokol generátor, obsluha rozhraní UART a zpracování části ladících komponent. Vzhledem k nutnosti zajištění generování zpráv v předem definovaném časovém intervalu bylo nutné většinu ostatních obslužných rutin optimalizovat z hlediska požadovaného výpočetního času. A to například u reakce na stisknutí tlačítek, kdy v obslužné rutině dochází pouze k nastavení příznaku, který je dále zpracován v hlavní smyčce.

Na nejnižší úrovni je zpracovávána hlavní smyčka, která zajišťuje zprostředkování informací uživateli ve formě vykreslování dat na displej, reakci na stisknutá tlačítka (bez nutnosti vysoké četnosti kontrol GPIO pinů, díky využití přerušení), zpracování a vysílání dat na UART a zpracovávání zbývajících částí ladících komponent.



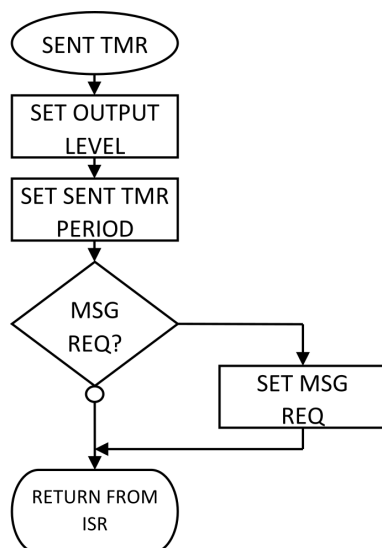
Obr. 25: Vývojový diagram se základním rozložením architektury FW

4.3.2 SENT protokol

Zásadní komponentu z hlediska funkcionality zařízení tvoří SENT protokol generátor. Data se přenáší po nibblech a přenášená hodnota je definovaná celkovou dobou mezi dvěma sestupnými hranami. Vzhledem k širokému spektru využívaných hodnot „tick time“ v rozsahu 3 až 90 μs byl pro tento účel využit časovač. V mikrokontroléru byl využit osmi bitový časovač s konfigurovatelnou pe-

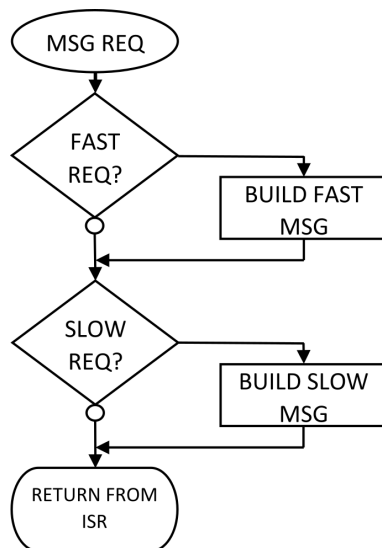
riodou. Jako zdroj hodinového signálu byl využit signál F_{OSC} , který je definován frekvencí krystalu 16 MHz násobenou pomocí PLL (x4) na hodnotu 64 MHz. Z hodinového signálu F_{OSC} byl pomocí „prescaler“ - předděličky definován signál 1 MHz pro časovač. Tím bylo definováno inkrementování hodnoty registru časovače každou $1 \mu s$. Načtením hodnoty do registru pro nastavení periody je možné definovat šířku pulsu na libovolnou hodnotu z rozsahu 1 - 256 μs .

Pro samotné generování signálu je tedy nutné mít hodnotu v μs s šířkou úrovně log.0 a log.1.



Obr. 26: Vývojový diagram pro rutinu přerušení zajišťující generování SENT protokolu

Komponenta generující hodnoty μs pro přenos zajistí výpočet potřebných dat na základě požadavku od SENT protokol generátoru. SENT protokol generátor vydává požadavky na generování zprávy rychlého kanálu a pomalého kanálu dle dokončení přenosu aktuální zprávy. Příprava dat se provádí na základě vybraného senzoru, respektive jeho komunikačního protokolu.



Obr. 27: Vývojový diagram pro rutinu přerušení zajišťující přípravu zpráv pro odeslání

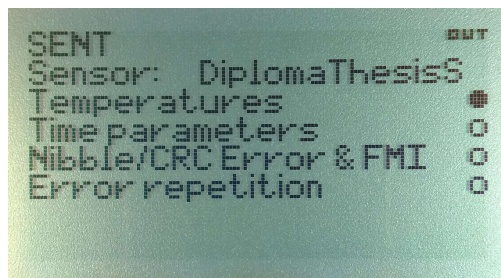
Využitý princip pro generování rozhraní SENT umožňuje generování signálu i s nejkratší hodnotou „tick time“. U simulátoru vzniklo z hlediska generování zpráv omezení v případě, kdy by nebylo využito fixní délky zpráv, tedy když by došlo k přeskočení pause pulsu. V případě tohoto by docházelo ke generování zpráv o délce 154 – 270 „ticků“. Vzhledem k tomu by docházelo ke generování nejkratších rámců s délkou 462 μs při hodnotě „tick time“ 3 μs . To by z hlediska výpočetního času nebylo možné s jistotou dodržet, jelikož samotný výpočet jedné rychlé zprávy s modifikovanou hodnotou „tick time“ vyžaduje přibližně 300 μs . V případě hodnoty „tick time“ 3 μs by došlo k využití více než 64 % výpočetního času na počítání rychlých zpráv, čímž by nezbyl čas pro výpočet pomalých zpráv a obsluhu zbytku zařízení. V případě modifikace „tick time“ na hodnotu 2,4 μs což odpovídá povolenému limitu -20 % z hodnoty „tick time“ by délka zprávy klesla na hodnotu 369 μs , kdy již není možné zajistit standardní výpočet rychlé zprávy včetně přepínání kontextu. V rámci zajištění, alespoň omezené funkcionality i v tomto nestandardním případě dochází k vyřazení proměnné délky „tick time“ to zapříčiní zkrácení doby výpočtu rychlé zprávy na hodnotu přibližně 180 μs . Při využití nejkratší možné fixní délky zpráv, tedy kdy každá zpráva trvá 282 „ticků“, je minimální délka jedné zprávy 846 μs . Při takto dlouhé zprávě je pro výpočet všech zpráv využito v průměru 41 % výpočetního času.

4.3.3 GUI

Grafické uživatelské rozhraní umožňuje konfiguraci všech nutných parametrů. Cele uživatelské rozhraní je uspořádáno do jednotlivých kategorií.

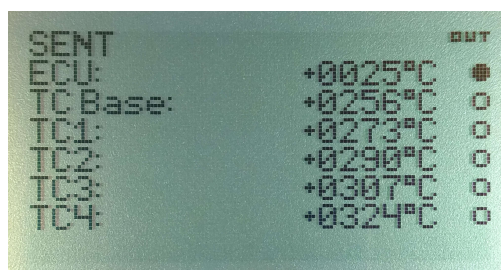
Úvodní menu zobrazuje jednotlivé kategorie do kterých jsou uspořádány parametry a název aktu-

álního senzoru.



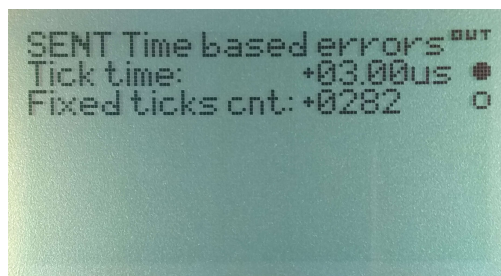
Obr. 28: Grafické uživatelské rozhraní - úvodní obrazovka

V rámci menu pro nastavení teplot je možné konfigurovat teplotu řídicí jednotky (ECU). Dále je možné konfigurovat základní společnou teplotu pro všechny termočlánky a odchylky pro jednotlivé termočlánky. Modifikace hodnot je možné provádět po jednotlivých řádech - změna o 1, 10, 100 či 1000.



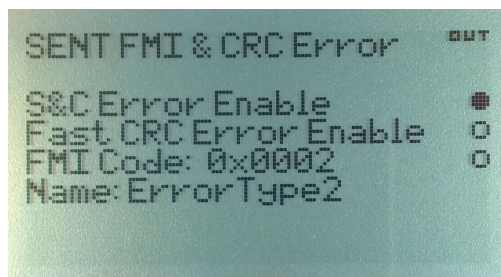
Obr. 29: Grafické uživatelské rozhraní - nastavení teplot

V dalším menu je možné konfigurovat časové parametry aktuálního senzoru. Je možné modifikovat hodnotu „Tick Time“, a to vždy o 3,3 % základní hodnoty.



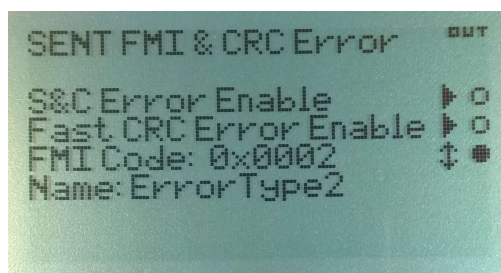
Obr. 30: Grafické uživatelské rozhraní - nastavení časů

V tomto menu je umožněno konfigurovat jednotlivé FMI kódy, aktivovat chybu CRC v rychlé zprávě a aktivovat chybu „Status&Communication“ nibblu. Pro přehlednost je v případě FMI zobrazen jak kódové označení, tak název daného FMI.



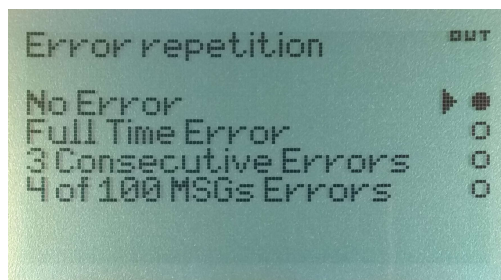
Obr. 31: Grafické uživatelské rozhraní - nastavení FMI kódů, chyby CRC a chyby S&C nibblu

Na obrázku 32 jsou zobrazeny aktivní chybové stavy. V případě aktivní chyby je zobrazena symbol šipky a v případě aktivního výběru FMI kódu je zobrazen symbol šipky symbolizující možný pohyb ve výběru FMI kódů.



Obr. 32: Grafické uživatelské rozhraní - nastavení FMI kódů, chyby CRC a chyby S&C nibblu - aktivní chyby

V posledním menu je možné zvolit opakování chyb. A to ve čtyřech variantách. Je možné chyby úplně deaktivovat, nastavit chybu do všech zpráv, či využít jeden z módů definovaných normou. Konkrétně se jedná o 3 po sobě jdoucí chybové rámce a nebo 4 nesouvislé chybové rámce.

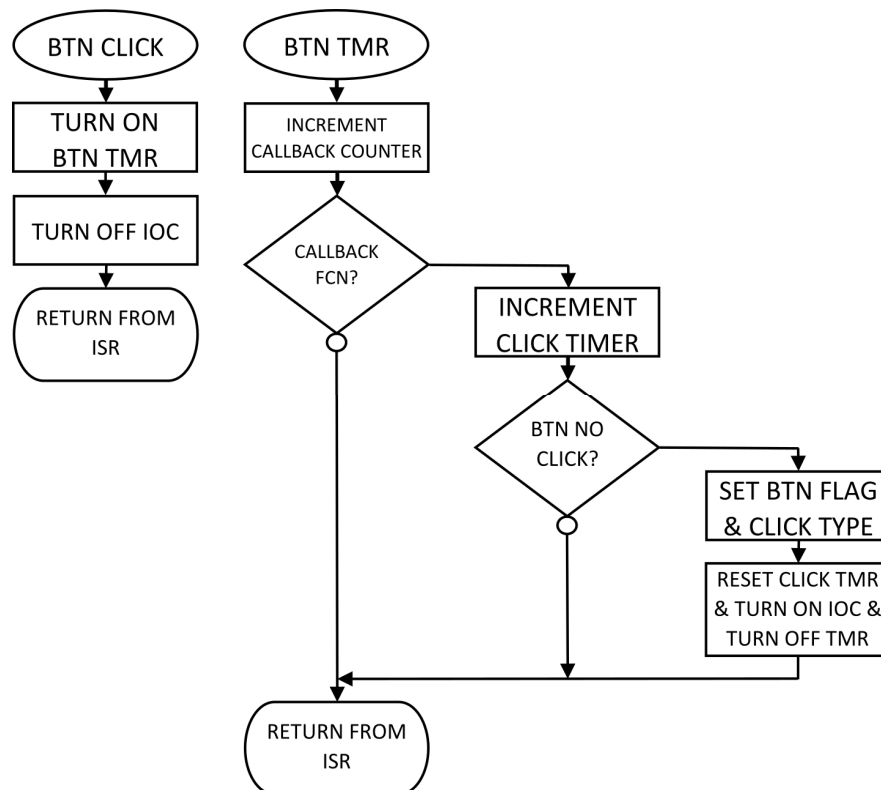


Obr. 33: Grafické uživatelské rozhraní - nastavení simulace opakování chyb

4.3.4 Ovládací prvky

Pro detekci stisku tlačítek, sloužících pro celkové ovládání simulátoru byl využit přerušovací systém. Konkrétně bylo využito dvou různých HW periférií a přerušení. Samotný stisk tlačítka je detekován pomocí IOC, které detekuje hrany na GPIO pinech MCU. IOC dokáže reagovat na náběžné i sestupné hrany signálu, ovšem vzhledem k HW konfiguraci simulátoru je potřeba detekovat pouze sestupné hrany na GPIO pinech. Po stisku tlačítka je vyvolána rutina pro obsluhu přerušení od zdroje IOC, ta deaktivuje IOC pro všechny tlačítka a aktivuje časovač.

Časovač aktivovaný obslužnou rutinou pro IOC zajišťuje dvojí funkcionalitu. První funkcí je tzv. „debouncing“, tedy ochrana proti zákmitům při stisku tlačítka jelikož se jedná o mechanický kontakt, které trpí touto vlastností. Druhou funkcí tohoto časovače je rozlišení délky stisku tlačítka. Obslužná rutina pro přerušení od časovače po definovaném počtu iterací volá „CallBack“ funkci (vždy po 100ms), která má za úkol počítat délku stisku tlačítka a na základě definované délky pro krátký a dlouhý stisk tlačítka vystavit příznak, který je následně zpracován v hlavní smyčce. Toto řešení je zvoleno jelikož se jedná o nenáročný způsob pro detekci stisku tlačítek z hlediska potřebného výpočetního času.



Obr. 34: Vývojový diagram pro rutinu přerušení zajišťující detekci stisku tlačítek

4.3.5 Grafický displej

Pro poskytování informací uživateli simulátoru a jeho konfiguraci byl využit grafický displej s řadičem ST7535. Jedná se o monochromatický grafický displej s rozlišení 128x64 pixelů. Pro komunikaci s grafickým řadičem displeje bylo využito SPI s přenosovou rychlostí 2,4 Mbit/s. Při standardním využití SPI je možné komunikovat mezi master a slave zařízení obousměrně, ovšem v případě využití displeje je možné komunikovat pouze směrem od master zařízení k slave zařízení. Z toho důvodu bylo nutné využít část paměti RAM v mikrokontroléru jako grafické paměti (z celkové paměti RAM 8kB bylo využito jako grafické paměti 1kB).

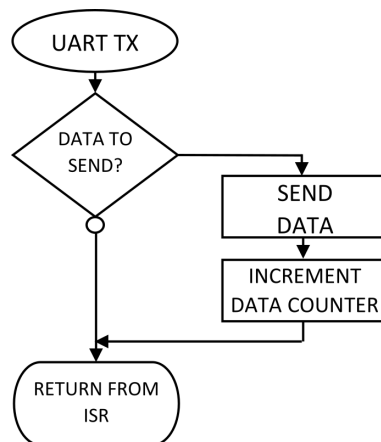
V rámci zajištění funkcionality displeje simulátoru bylo nutné navrhnout a naprogramovat grafickou knihovnu, která zajišťuje veškerou následně využívanou funkcionalitu. Navržená grafická knihovna zajišťuje vykreslování jednotlivých znaků z ASCII tabulky, vykreslování vlastních znaků s omezením na velikost použitého fontu 5x7 (respektive 5x8) pixelů, vykreslování textových řetězců, vykreslování čar, obdélníků, rámečků a obrázků (bitmapa).

Zápis na displej se provádí tak, že je nejdříve nutné volat část API, které zajistí požadované vykreslení do grafické paměti, a následně je nutné volat z API obnovení displeje. API podporuje dvě varianty pro obnovení obrazu displeje, první variantou je obnovení celého displeje a druhou variantou je obnovení části displeje. V případě obnovování pouze části displeje je nutné zvolit dvojici souřadnic v rozměru X a Y. Souřadnice jsou funkcí pro obnovení analyzovány a je přepočítána minimální oblast, kterou je možné obnovit, tak aby bylo zajištěno obnovení požadované oblasti. Toto je nutné jelikož není možné obnovovat jednotlivé bity. Matice displeje obsahuje 128x64 pixelů. Tato matice je rozdělena na 128 sloupců a 8 řádků. Vzhledem k omezení ve smyslu 8 řádků je nutná zmiňovaná analýza a přepočet minimální oblasti pro obnovení, aby při využívání API nebylo nutné uvažovat řádkování displeje.

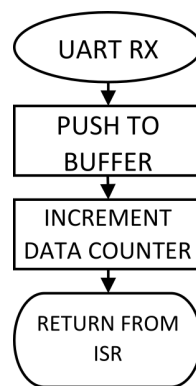
4.3.6 UART

V rámci zajištění komunikace s počítačem bylo využito rozhraní UART. Pro komunikaci bylo využito dvou zdrojů přerušení, a to od vysílacího a přijímacího kanálu ve spolupráci s dvojicí bufferů. Pro přístup k periférii je využito API, které v případě vysílání zajistí zápis do příslušného bufferu a povolení patřičného zdroje pro systém přerušení. Po vyslání bajtu dat dojde k vyvolání rutiny pro obsluhu přerušení od vysílacího kanálu, v rámci této rutiny je provedena kontrola vysílacího bufferu a v případě dostupných dat pro odeslání se provede zápis do vysílacího registru UARTu. Pokud nejsou dostupná další data k odeslání, dochází k zakázání přerušení od vysílacího kanálu.

U přijímacího kanálu je trvale aktivní přerušení. V případě příchozího bajtu zajistí obslužná rutina přerušení přenos dat z přijímacího registru UARTu do příslušného bufferu.



Obr. 35: Vývojový diagram pro rutinu přerušení zajišťující obsluhu vysílání na UART



Obr. 36: Vývojový diagram pro rutinu přerušení zajišťující obsluhu příjmu na UART

5 Možnosti komunikace s PC

V rámci budoucích rozšíření tohoto simulátoru je uvažováno o variantě ovládání simulátoru pomocí PC. Pro zajištění takové komunikace se předpokládá využití rozhraní UART na straně mikrokontroléru a virtuálního COM portu na straně počítače.

Na úrovni HW je připraveno komunikační rozhraní. Ze strany FW v mikrokontroléru je příprava na možnou implementaci této části obsluhy simulátoru. Je zajištěna přístupnost konfigurovatelných parametrů a API pro přístup k rozhraní UART.

5.1 Návrh komunikačního protokolu

V rámci navrženého komunikačního protokolu bylo nutné taktéž definovat symboly pro začátek zprávy, konec zprávy a escape sekvence pro znaky využité jako začátek a konec zprávy. Tyto symboly a escape sekvence jsou definovány v tabulce 12.

Tab. 12: Definice symbolů SOF, EOF, escape sekvence

Start of Frame (SOF)	0x0102
End of Frame (EOF)	0x0304
0x01	0x0101
0x03	0x0303

Kompletní navržený komunikační protokol se nachází v přílohách, konkrétně jako příloha B. V rámci komunikačního protokolu jsou definovány tři typy zpráv. První typ zpráv slouží pro nahrávání parametrů specifikujících senzor. Druhý typ zpráv slouží pro ovládání simulátoru. A třetí typ zpráv je využit pro odpovědi od simulátoru.

Tab. 13: Typy příkazů v navrženém komunikačním protokolu

Typ zprávy	CMD (XX = 00 - FF)	DATA
Příkazy pro nahrávání specifikace senzoru	0xDDXX	DATA
Příkazy pro ovládání simulátoru	0xCCXX	DATA
Odpovědi od simulátoru	0xCEXX	DATA

Tab. 14: Příklady jednotlivých rámců

	SOF	CMD	DATA	EOF
Nastavení jména senzoru	0x0102	0xDD00	"DiplomaThesisS"	0x0304
Nastavení typu senzoru \$048	0x0102	0xDD01	0x48	0x0304
Povolení výstupu	0x0102	0xCC00		0x0304
OK Odpověď na žádost o povolení výstupu	0x0102	0xCE00	0xCC00	0x0304

6 Testování vyvinutého zařízení

6.1 Popis možného senzoru

Pro účely testování byl navržen senzor typu \$048[6]. Norma tento typ senzoru specifikuje jako vysoko teplotní senzor s multiplexovaným rychlým kanálem. Norma dále specifikuje rozložení dat v rámci.

Tab. 15: Specifikace senzoru \$048

Nibble					
1	2	3	4	5	6
FC	DCC	MSN	MidSN	LSN	Zeros

Pro přenos dat jsou v tomto typu senzoru vyhrazeny 3 nibbly v rychlém rámci, tedy hodnota s rozsahem 12 bitů. Jedná se o polovinu původního dostupného rozsahu, ovšem díky multiplexování rychlého rámce je možné takto přenést až 16 různých rychlých rámců.

V tomto konkrétním případě bylo využito pouze 4 rychlých rámců z možných 16, jelikož senzor posílá 4 teploty.

Pro rychlé rámce je definována pevná délka rámce 282 „ticků“ a „tick time“ $3 \mu s$, z toho vyplývá délka jednoho rychlého rámce $846 \mu s$. Pro převod teplot byla zvolena převodní křivka pro TC i ECU definovaná normou SAE J2716-2016, strana 83.

Pro pomalý kanál byl zvolen typ „Enhanced“, který umožňuje přenos 8 bitového ID a 12 bitových dat. Pro FMI a stavové kódy bylo využito ID 0x01, pro přenos teploty ECU bylo využito ID 0x23 - jedno z možných ID definovaných normou pro doplňující data ze senzoru.

Tab. 16: Specifikace navrženého senzoru pro účely této diplomové práce - část 1

Parametr	Hodnota
TickTime	3 μs
Typ rychlého kanálu	\$048
Typ pomalého kanálu	Enhanced
Pevná délka rámce	282 ticks
TC1	ANO
TC2	ANO
TC3	ANO
TC4	ANO
Teplotní křivka TC - Offset	73
Teplotní křivka TC - sklon	3
Teplotní křivka ECU - Offset	73
Teplotní křivka ECU - sklon	3

Tab. 17: Specifikace navrženého senzoru pro účely této diplomové práce - část 2

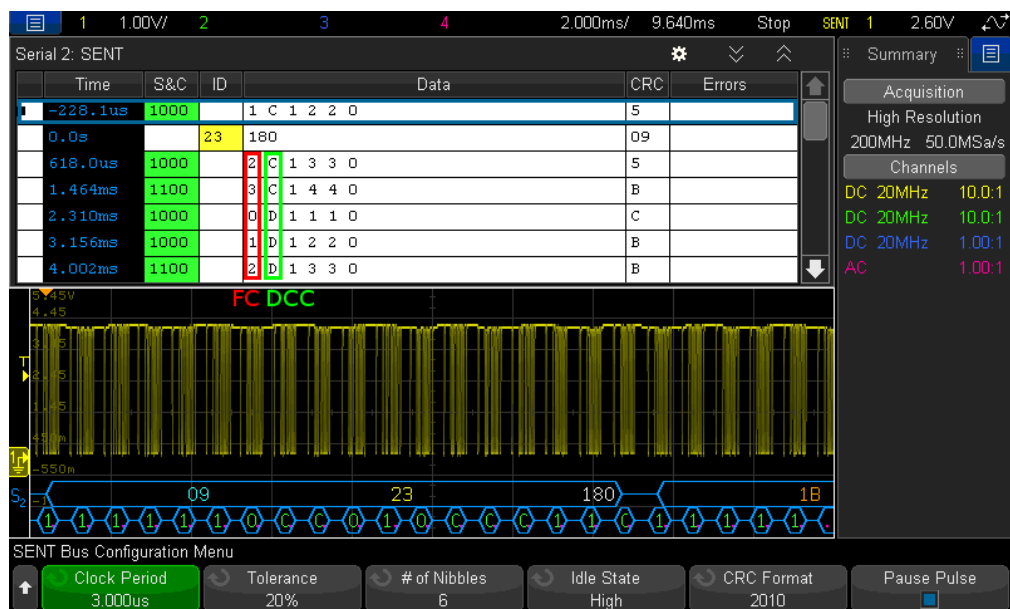
ID - pomalý kanál	Název FMI	Data - rychlý kanál	Data - pomalý kanál
0x23	NoError	Beze změny	0x000
0x23	ErrorType1	0xFF1	0x001
0x23	ErrorType2	0xFF2	0x002
0x23	ErrorType3	0xFF3	0x003

Tab. 18: Specifikace navrženého senzoru pro účely této diplomové práce - část 3

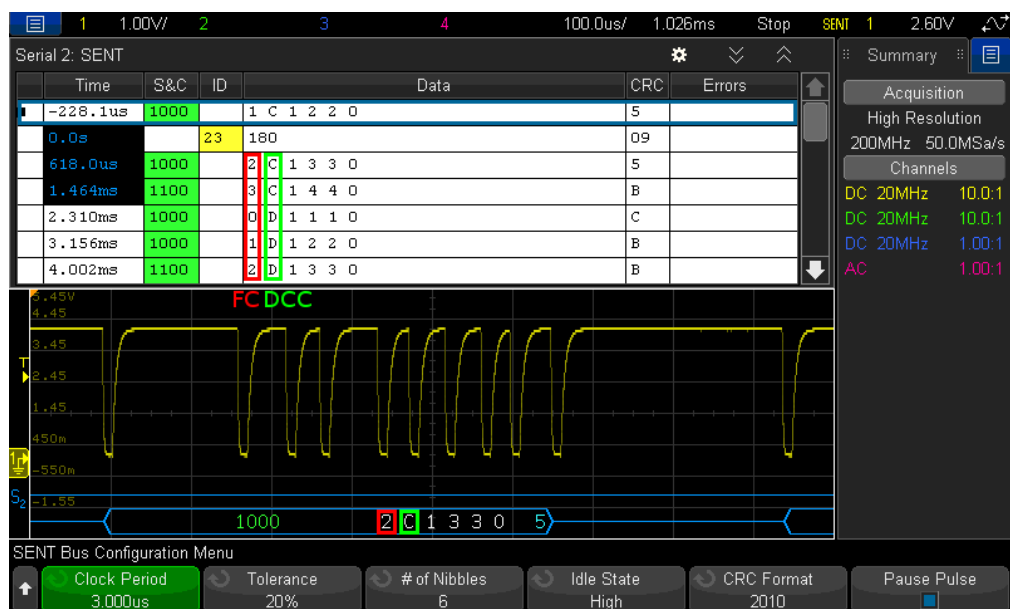
ID - pomalý kanál	Obsah zprávy	Data - rychlý kanál	Data - pomalý kanál
0x01	Status & FMI	dle tab. 17	dle tab. 17
0x23	Teplota ECU	Beze změny	Teplota ECU

6.2 Verifikace funkčnosti SENT rozhraní

V rámci testování SENT rozhraní bylo využito navrženého senzoru dle specifikace v kapitole 6.1. Navržený senzor byl využit pro ověření všech typů senzoru, které tento simulátor podporuje. V případě typů senzorů bez multiplexování rychlého kanálu byla odvozena redukovaná verze ze specifikovaného typu \$048.



Obr. 37: Dekódované rámce pomocí osciloskopu, aktivní multiplexování rychlého kanálu (FC, DCC)



Obr. 38: Dekódovaný rámec pomocí osciloskopu, aktivní multiplexování rychlého kanálu (FC, DCC)

Tab. 19: Dekódované rámce vyslané simulátor konfigurovaným dle specifikace senzoru v kapitole 6.1

RAW data	Serial bit 3	Serial bit 2	FC	DCC	Data	CRC	Slow Channel
0x81310006	1	0	0x1	0x3	0x100	0x6	ID - 0x23; Data - 0x180; CRC - 0x09
0x82310003	1	0	0x2	0x3	0x100	0x3	
0xC331000B	1	1	0x3	0x3	0x100	0xB	
0x80410003	1	0	0x0	0x4	0x100	0x3	
0x8141000B	1	0	0x1	0x4	0x100	0xB	
0xC241000E	1	1	0x2	0x4	0x100	0xE	
0x03410006	0	0	0x3	0x4	0x100	0x6	
0x0051000D	0	0	0x0	0x5	0x100	0xD	
0x01510005	0	0	0x1	0x5	0x100	0x5	
0x42510000	0	1	0x2	0x5	0x100	0x0	
0xC3510008	1	1	0x3	0x5	0x100	0x8	
0x00610002	0	0	0x0	0x6	0x100	0x2	
0x0161000A	0	0	0x1	0x6	0x100	0xA	
0x0261000F	0	0	0x2	0x6	0x100	0xF	
0x03610007	0	0	0x3	0x6	0x100	0x7	
0x8071000C	1	0	0x0	0x7	0x100	0xC	
0x81710004	1	0	0x1	0x7	0x100	0x4	
0x02710001	0	0	0x2	0x7	0x100	0x1	
Začátek pomalé zprávy							
ID pomalé zprávy							
Data pomalé zprávy							
CRC pomalé zprávy							

6.3 Generování chyb na rozhraní SENT

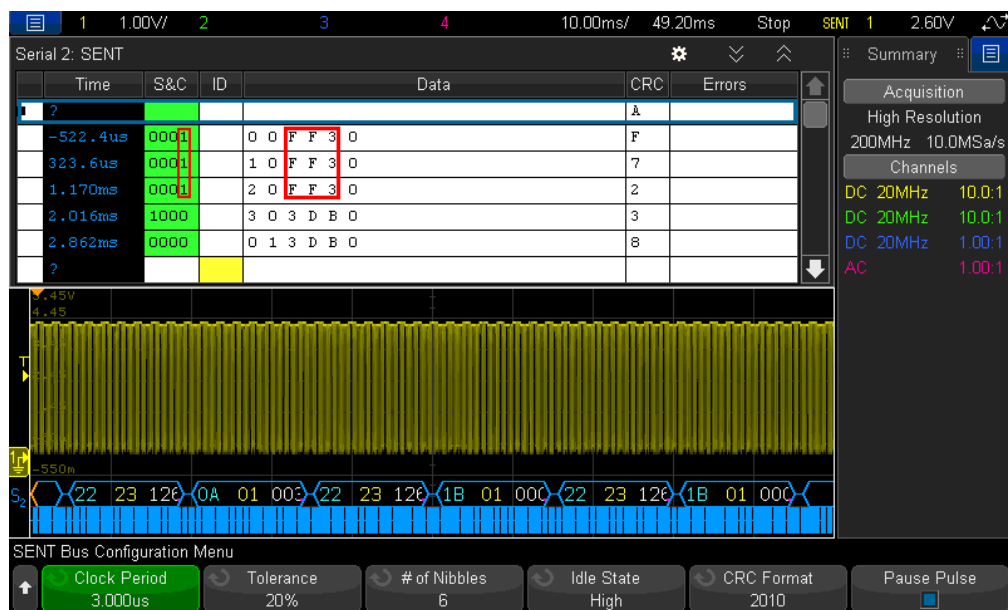
V rámci požadavků na simulátor bylo definována možnost zasáhnutí do časových parametrů simulovaného senzoru. To je možné pomocí modifikace hodnoty „tick time“. Norma připouští toleranci z dlouhodobého hlediska až $\pm 20\%$. Simulátor umožňuje provést změnu hodnoty „tick time“ v $\approx 3,3\%$ krocích s možností až 10 kroků oběma směry. V tabulce 20 se nachází výsledky měření z verifikace funkčnosti tohoto krokování. V případě měření pro 6 a 12 μs skutečné hodnoty odpovídají s dobrou přesností. V případě 3 μs jsou odchylky větší, ovšem pro zajištění vyšší přesnosti by bylo nutné modifikovat rutinu, která zajišťuje výpočet. To by vedlo k neúměrnému navýšení potřebného výpočetního času z důvodu komplikovanějších výpočtů s plovoucí desetinou čárkou pro kterou zvolený

mikrokontrolér nemá HW podporu.

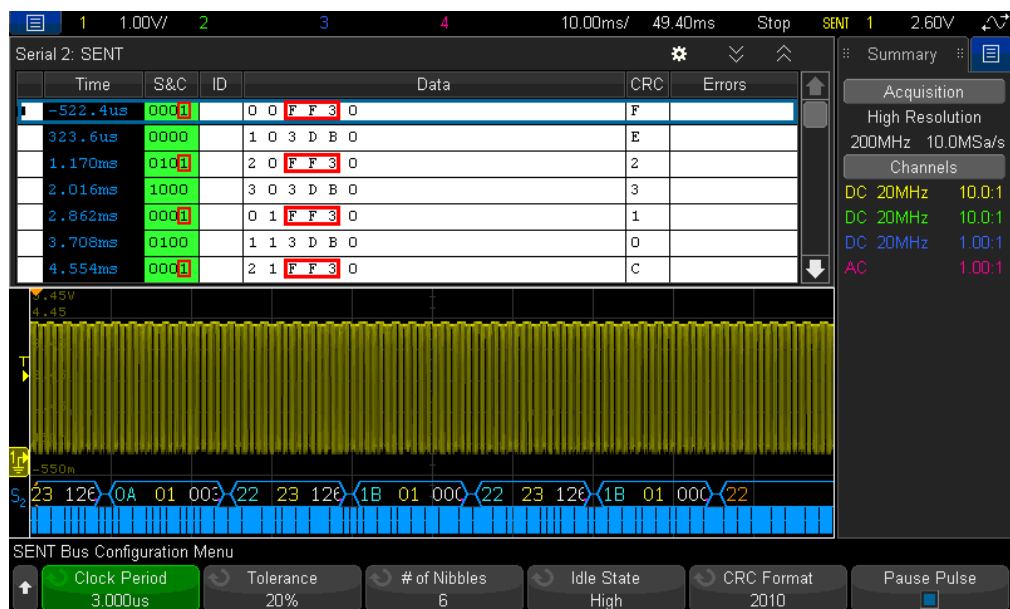
Tab. 20: Verifikace modifikovaných hodnot tick time, 12 μs , 6 μs a 3 μs

-	Nastavená hodnota tick time μs	Měřená délka Sync pulsu μs	Vypočítaná hodnota tick time μs
12 μs			
-6	9,6	538	9,61
-5	10,0	560	10,00
-4	10,4	580	10,36
-3	10,8	605	10,80
-2	11,2	625	11,16
-1	11,6	650	11,61
0	12,0	672	12,00
1	12,4	692	12,36
2	12,8	717	12,80
3	13,2	737	13,16
4	13,6	762	13,61
5	14,0	784	14,00
6	14,4	804	14,36
6 μs			
-6	4,8	269	4,80
-5	5	280	5,00
-4	5,2	289	5,16
-3	5,4	300	5,36
-2	5,6	314	5,61
-1	5,8	325	5,80
0	6	336	6,00
1	6,2	345	6,16
2	6,4	356	6,36
3	6,6	370	6,61
4	6,8	381	6,80
5	7	392	7,00
6	7,2	401	7,16
3 μs			

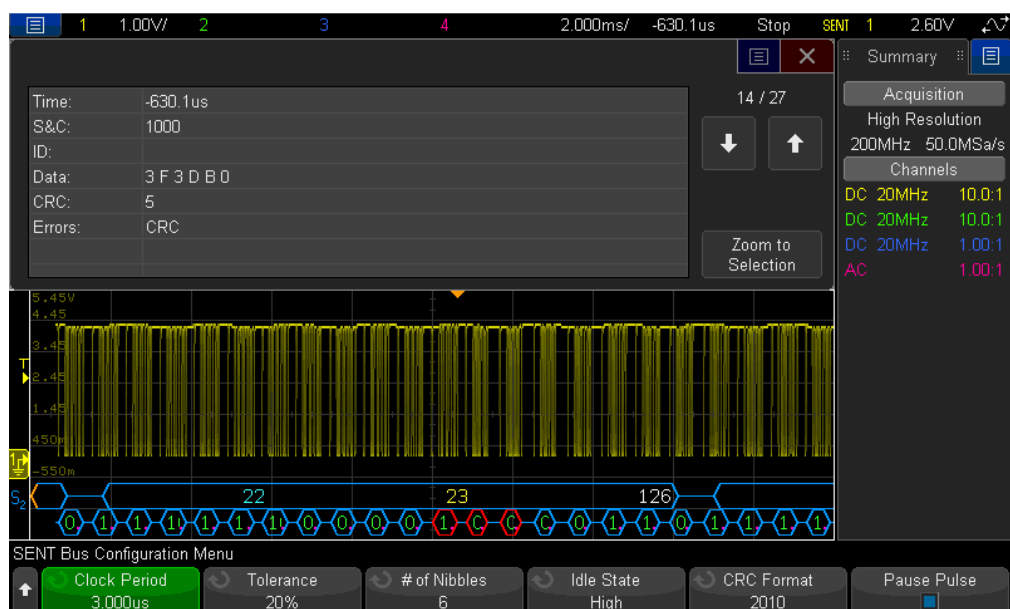
	Nastavená hodnota tick time	Měřená délka Sync pulsu	Vypočítaná hodnota tick time
-	μs	μs	μs
-6	2,4	-	-
-5	2,5	140	2,50
-4	2,6	146	2,61
-3	2,7	149	2,66
-2	2,8	157	2,80
-1	2,9	160	2,86
0	3	168	3,00
1	3,1	174	3,11
2	3,2	177	3,16
3	3,3	185	3,30
4	3,4	188	3,36
5	3,5	196	3,50
6	3,6	-	-



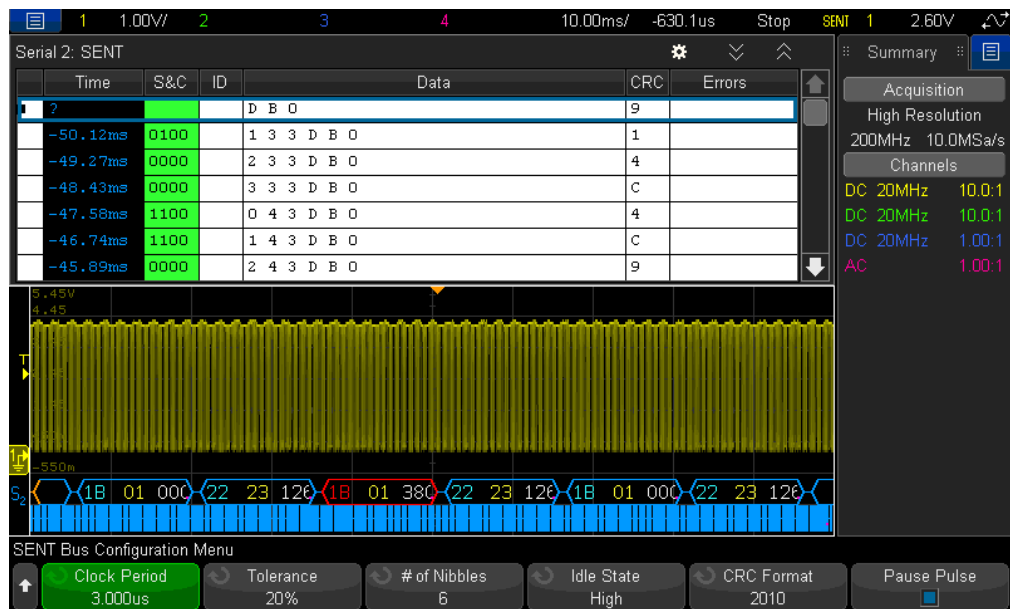
Obr. 39: Simulace tří souvislých chybových rámců



Obr. 40: Simulace čtyř nesouvislých chybových rámců



Obr. 41: Simulace chyby CRC rychlého rámce



Obr. 42: Simulace chyby S&C nibblu -> chyba pomalého rámce

7 Zhodnocení výsledků a závěr

Ústředním tématem této práce bylo navržení a realizace systému, zajišťujícího možnost přesně a definovaně simulovat poruchové stavy, ke kterým může u senzorů docházet. V úvodních kapitolách byla rozebrána problematika sběrnic a rozhraní využívaných pro automobilový průmysl. Komunikační sběrnice a rozhraní pro rozbor byly zvoleny s ohledem na vyvíjená řešení společností Continental Powertrain Czech Republic s.r.o. a aplikace vysokoteplotních senzorů. S ohledem na zadání práce byl důraz kladen primárně na komunikační rozhraní SENT, dle nejnovější revidované verze normy SAE J2716-2016, jakožto komunikačního rozhraní využitého v realizovaném simulátoru.

V rámci teoretické části této diplomové práce byl proveden rozbor možných poruchových stavů, ke kterým může u senzorů dojít. Z hlediska možných příčin vzniku těchto chyb byla uvedena návaznost na problematiku testování v oblasti EMC.

Za účelem splnění cíle této práce byl zpracován návrh HW a FW architektury simulátoru. Na základě tohoto návrhu bylo zpracováno obvodové řešení, které bylo následně ověřeno v rámci simulací jednotlivých funkčních bloků. Po prověření jednotlivých bloků byl navržen layout plošného spoje.

Pro realizovaný a oživený HW simulátoru byl dle předchozího návrhu FW architektury implementován program s využitím předem připravených FW komponent. FW simulátoru umožňuje simulovat sadu typů senzorů definovaných normou na základě specifikace simulovaného senzoru. Je možné využít oba formáty pomalé zprávy, které jsou připouštěné normou.

FW komponenta zabezpečující funkcionalitu SENT rozhraní umožňuje věrně simulovat senzor podle zadané specifikace. V rámci specifikace senzoru je možné definovat kompletní chování simulovaného senzoru. Je možné definovat teplotní parametry ve smyslu definování převodní křivky, rozsahu teplot a aktivních termočlánků. Dále je možné definovat časové parametry, konkrétně „tick time“ a pevnou délku zpráv. Dále je možné definovat libovolné pomalé zprávy a jejich cyklus opakování.

V rámci definování pomalých zpráv je možné v tuto chvíli specifikovat až 32 zpráv, kdy je každá zpráva složena z 8 bitového ID a 12 bitů dat. Mezi pomalé zprávy patří stavové hlášení, respektive hlášení poruchových stavů (FMI). Pro toto hlášení je definována další řada kódů definující jednotlivé chybové hlášení. Pro každé chybové hlášení je možné také pevně definovat vysílanou hodnotu v rychlých zprávách po dobu trvání chybového stavu.

Z hlediska simulace chyb je možné simulovat chyby v datech ve smyslu prudkých změn v teplotách. Dále simulátor umožňuje simulovat chyby ve „S&C“ nibblu, což způsobí chybu pomalé zprávy. Také je možné simulovat chyby v CRC rychlých zpráv.

Pro zajištění potřebné funkcionality je možné taktéž simulovat opakování poruchových stavů, které definuje norma pro testy z oblasti EMC. Je možné simulovat 3 souvislé chybové rámce, 4 nesouvislé chybové rámce ze 100 a trvalou chybu.

Výše jmenovanou funkcionalitou bylo kompletně naplněno zadání této diplomové práce, došlo k na-

plnění veškerých původních požadavků, které byly na simulátor kladeny.

Možná rozšíření simulátoru:

Nejpodstatnějším rozšíření které se do budoucna uvažuje je implementace komunikace s PC dle návrhu v kapitole 5 a komunikačního protokolu - Příloha B. Pro plnohodnotné využití se předpokládá taktéž návrh a realizace aplikačního SW na PC, který bude zajišťovat ovládání simulátoru, včetně nahrávání konfigurací senzorů.

Dalším případným rozšíření by mohla být implementace nových typů senzorů pro generátor zpráv na rozhraní SENT. Z hlediska stávajícího FW je zajištěna snadná implementace nových typů senzorů s napojením na stávající FW struktury.

Literatura

- [1] *ISO 11898-1: Road vehicles - Controller area network (CAN) - Part 1: Data link layer and physical signalling*. Second edition. Switzerland: International Organization for Standardization, 2015.
- [2] *ISO 11898-2: Road vehicles - Controller area network (CAN) - Part 2: High speed medium access unit*. Second edition. Switzerland: International Organization for Standardization, 2016.
- [3] *ISO 17987-3: Road vehicles - Local Interconnect Network (LIN) - Part 3: Protocol specification*. First edition. Switzerland: International Organization for Standardization, 2016.
- [4] *PSI5: Peripheral Sensor Interface – Base Standard*. V2.3. PSI5 Organization, 2018.
- [5] PSI5 Organization. Organization. *PSI5*, [online]. [cit. 2018-09-30] Dostupné z: <http://www.psi5.org>.
- [6] *SAE J2716: SENT - Single Edge Nibble Transmission for Automotive Applications*. APR2016. Warrendale: SAE International, 2016.

Seznam příloh

- Příloha A: Specifikace navrženého senzoru - 2 strany
- Příloha B: Návrh komunikačního protokolu - 2 strany
- Příloha C: Osazovací plán simulátoru strana TOP - výkres A4, číslo výkresu: 2019011402
- Příloha D: Osazovací plán simulátoru strana BOT - výkres A4, číslo výkresu: 2019011401
- Příloha E: Schéma simulátoru - výkres A3, číslo výkresu: 2019011400
- Příloha F: FW Simulátoru - příloha na CD

Příloha A: Specifikace navrženého senzoru

Kód 1: Vytvořená specifikace senzoru pro účely této diplomové práce

```
/* SENSOR TYPE $048 - FAST CHANNEL*/
uint8_t  sensorType = 48;

/* ENHANCED SERIAL CHANNEL - SLOW CHANNEL */
uint8_t  sensorSlowChannelType = 0x02;

/* TickTime 3us */
uint8_t  sensorTickTime = 3;
int16_t  sensorTickTimeModifierMultiplier;
/* Step limit for SENT TickTime change */
int16_t  sensorTickTimeModifierMultiplierLimH = 10;
int16_t  sensorTickTimeModifierMultiplierLimL = -10;

/* Fixed message length set to 282 ticks */
uint16_t sensorFixedTicks = 282;
int16_t  sensorFixedTicksModifier;
int16_t  sensorFixedTicksModifierLimH = 0x7FFF;
int16_t  sensorFixedTicksModifierLimL = 0;

/* All TCs available */
uint8_t  sensorThermocoupleAvailable = 0b1111;

/* Temperature offset for each TC */
int16_t  sensorTCTemp[4] = {0x00, 0x00, 0x00, 0x00};
/* ECU (CJ) temperature & temperature limits*/
int16_t  sensorECUTemp = 0x019;
int16_t  sensorECUTempLimH = 437;
int16_t  sensorECUTempLimL = -72;
/* ECU temperature curve offset & multiplier - SAE J2716-2016 page 82/83
   */
int16_t  sensorECUTempOffset = 73;
int16_t  sensorECUTempMultiplier = 3;

/* Basic TCs temperature & temperature limits */
int16_t  sensorTCBaseTemp = 0x0100;
int16_t  sensorTCBaseTempLimH = 1289;
int16_t  sensorTCBaseTempLimL = -72;
```



```

int16_t  sensorTCTempLimH;
int16_t  sensorTCTempLimL;
/* TCs temperature curve offset & multiplier - SAE J2716-2016 page 82/83
*/
int16_t  sensorTCTempOffset = 73;
int16_t  sensorTCTempMultiplier = 3;

/* Active FMI code position in array: sensorFMIandStatusCodes */
uint16_t sensorFMICode = 2;
uint16_t sensorFMICodeLimH = SENSOR_FMI_MAX_COUNT;
uint16_t sensorFMICodeLimL = 0;
/* Available Status & FMI codes (0xFFEE - End of FMI list) */
uint16_t sensorFMIandStatusCodes[SENSOR_FMI_MAX_COUNT] =
    {0x000,0x001,0x002,0x003,0xFFEE};
/* Status & FMI codes names */
char
    sensorFMIandStatusCodesNames[SENSOR_FMI_MAX_COUNT][SENSOR_FMI_NAME_SIZE]
    = {"NoError"}, {"ErrorType1"}, {"ErrorType2"}, {"ErrorType3"};
/* Fast channel value during active FMI */
uint16_t sensorFMIandStatusCodesFastChanel[SENSOR_FMI_MAX_COUNT] =
    {SENSOR_FMI_FAST_MEAS_VAL, 0xFF1, 0xFF2, 0xFF3};
/* Cycle of serial messages specified in array: sensorSerialMessages */
uint8_t  sensorSerialMessageCycle[SENSOR_SERIAL_CYCLE_MAX_SIZE] =
    {0,1,SENSOR_SERIAL_CYCLE_NO_MSG};
/* Serial messages: IDs & Values */
uint16_t sensorSerialMessages[SENSOR_SERIAL_MESSAGES_MAX_COUNT][2] =
    {{0x01,SENSOR_FMI_SLOW_CODE},{0x23,SENSOR_FMI_FAST_MEAS_VAL}};
/* Set serial message ID which is used for Status & FMI codes */
uint8_t  sensorFMIID = 0x01;

/* Sensor name */
char  sensorName[15] = {"DiplomaThesisS"};

```

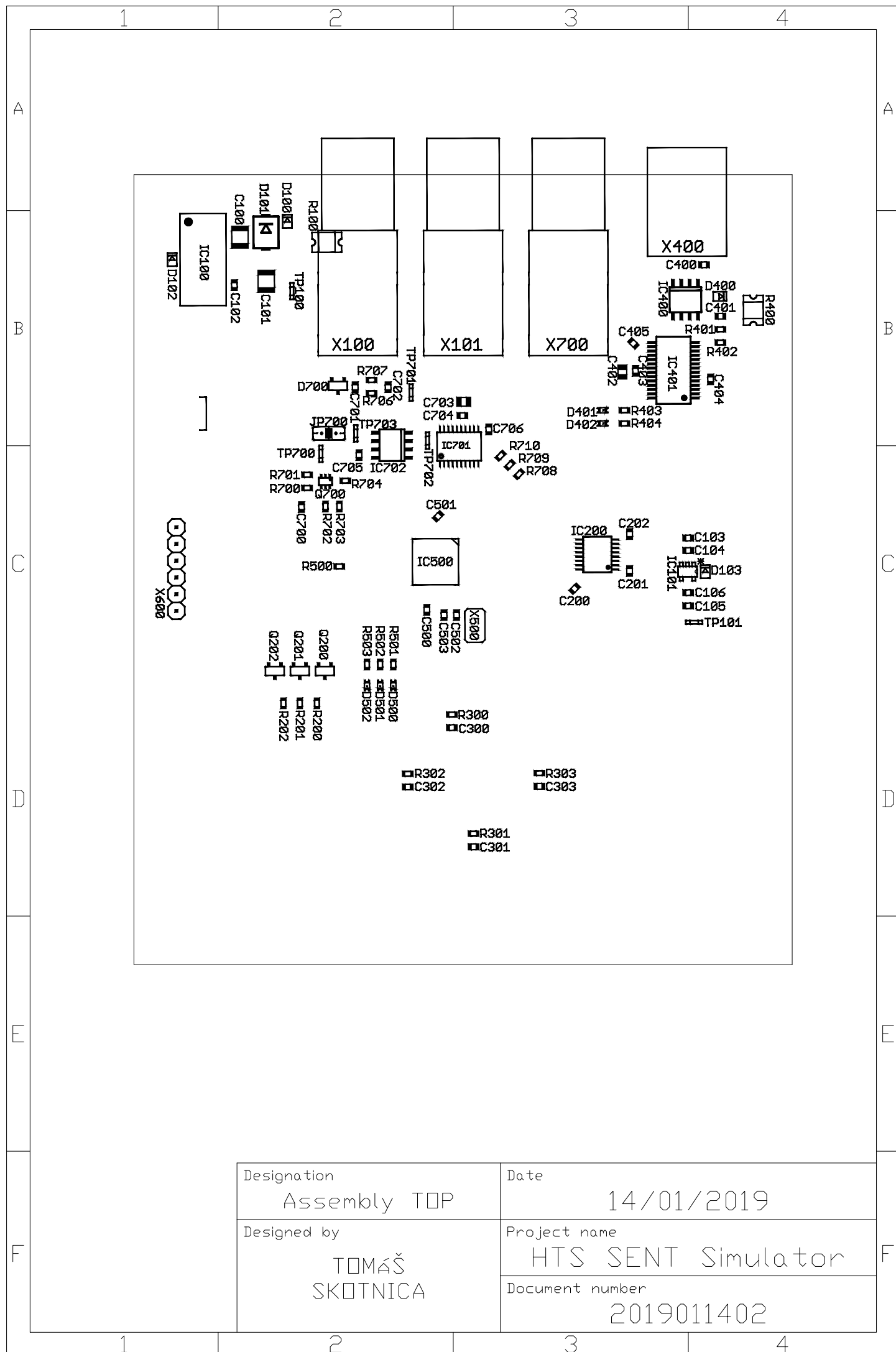
Příloha B: Návrh komunikačního protokolu

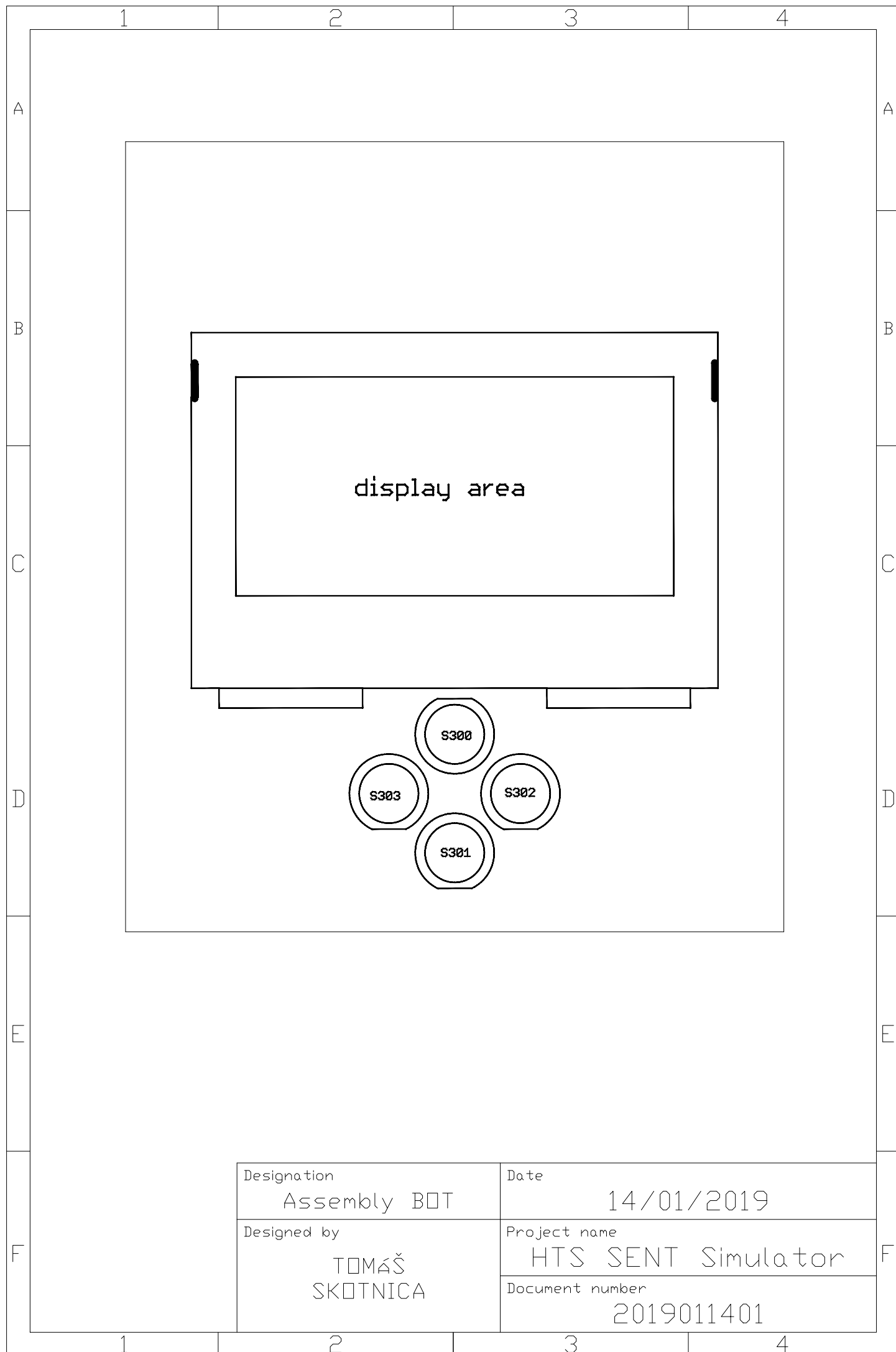
Sensor Specification	CMD	DATA (MSB first)
Sensor Name	0xDD00	Up to 14 Characters, Unused Fill with zeros
Sensor Type	0xDD01	Sensor type Number (45/46/48), uint8
sensor Slow Channel Type	0xDD02	Slow Channel Type (0 - None, 1 - Short, 2 Enhanced), uint8
Tick Time	0xDD03	Tick time value (3 - 90), uint8
Fixed Frame Legth (Ticks Count)	0xDD04	Fixed frame length, uint16 value
Thermocouple Availability	0xDD05	Available thermocouples (TC4-0x08; TC3-0x04; TC2-0x02; TC1-0x01; or its combination); uint8
TC Temp Curve Multiplier	0xDD06	uint8; Example: SAE J2716-2016, page 83, Value = 3
TC Temp Curve Offset Deg Celsius	0xDD07	int16; Example: SAE J2716-2016, page 83, Offset -200K -> to Celsius -> Value = 72
TC Temp Base Limit High	0xDD08	int16; Example: SAE J2716-2016, page 83, Maximal value 1562K -> to Celsius -> Value = 1289
TC Temp Base Limit Low	0xDD09	int16; Example: SAE J2716-2016, page 83, Minimal value 200.33K -> to Celsius -> Value = -72
ECU Temp Curve Multiplier	0xDD0A	uint8; Example: SAE J2716-2016, page 82, Value = 8
ECU Temp Curve Offset Deg Celsius	0xDD0B	int16; Example: SAE J2716-2016, page 82, Offset -200K -> to Celsius -> Value = 72
ECU Temp Limit High	0xDD0C	int16; Example: SAE J2716-2016, page 82, Maximal value 711K -> to Celsius -> Value = 437
ECU Temp Limit Low	0xDD0D	int16; Example: SAE J2716-2016, page 82, Minimal value 200.125K -> to Celsius -> Value = -72
FMI Message ID	0xDD0E	Serial message ID for FMI codes; uint8
FMI Code, FastChannel Value, Name	0xDD0F	FMI Code ID, uint8; Fast Channel Value (0x1000 - TC temp, other values are fixed), uint16; FMI Code Name, up to 16 Characters, unused fill with zeros

Sensor Specification	CMD	DATA (MSB first)
Serial Message ID, Data	0xDD10	ID, uint8; DATA, uint16 (0x1000 - ECU temp, 0x2000 - FMI Code ID from CMD 0xDD0E, other values are fixed, like SerialNumber, Manufacturer ID, etc.)
Serial Message Cycle	0xDD11	Cycling of Serial Messages - Set Serial message number in order to its addition, up to 31 x uint8, last value have to be 0xFF

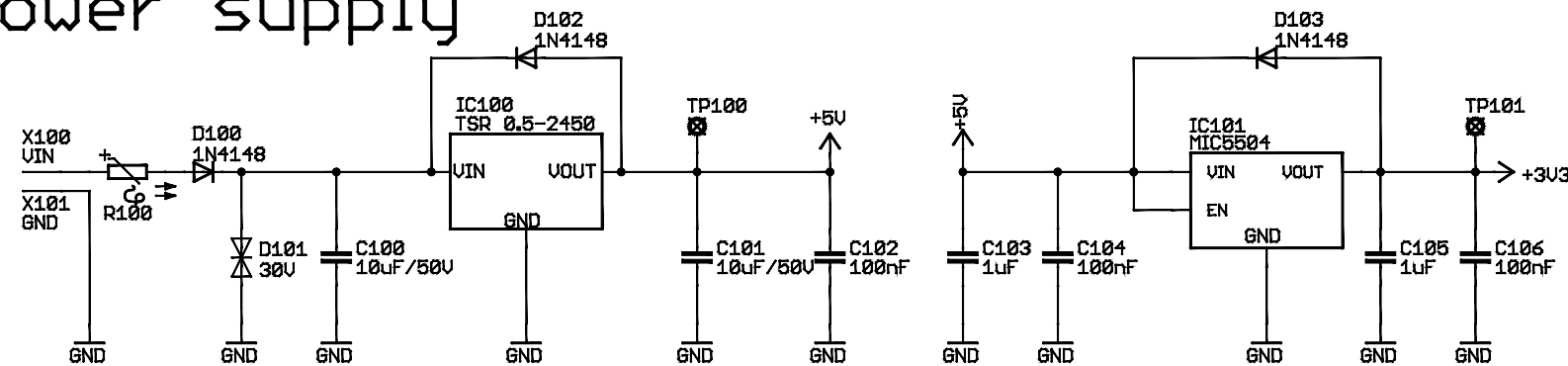
Simulator Control	CMD	DATA (MSB first)
Enable Output	0xCC00	no data
Disable Output	0xCC01	no data
Set TC Base Temp	0xCC02	int16, temperature in degrees Celsius
Set TC1 temp offset	0xCC03	int16, temperature in degrees Celsius
Set TC2 temp offset	0xCC04	int16, temperature in degrees Celsius
Set TC3 temp offset	0xCC05	int16, temperature in degrees Celsius
Set TC4 temp offset	0xCC06	int16, temperature in degrees Celsius
Set ECU Temp	0xCC07	int16, temperature in degrees Celsius
Set Tick Time Modifier	0xCC08	uint8, $\pm n \cdot 3,3\%$
Set Frame Length Modifier	0xCC09	int16, offset from base value (ticks)
Set FMI Code	0xCC0A	uint8
Set Error Repetition Type	0xCC0B	uint8, 0 - 3 consecutive error frames, 1 - 4 no consecutive error frames per 100 frames, 2 - full time error, 3 - no error
Get Parameter	0xCC0C	uint16, CMD which set required parameter

Simulator Response	CMD	DATA (MSB first)
OK	0xCE00	CMD for which one is this reply
NOK	0xCE01	CMD for which one is this reply
Get Parameter	0xCE02	int24, returned data type changed to int24

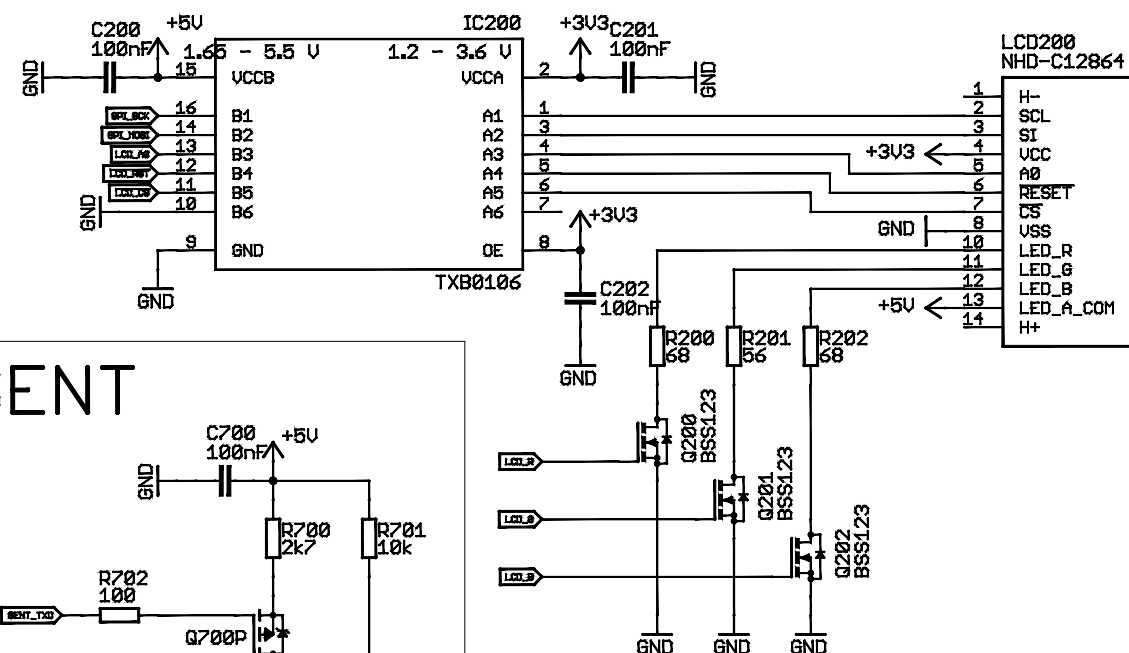




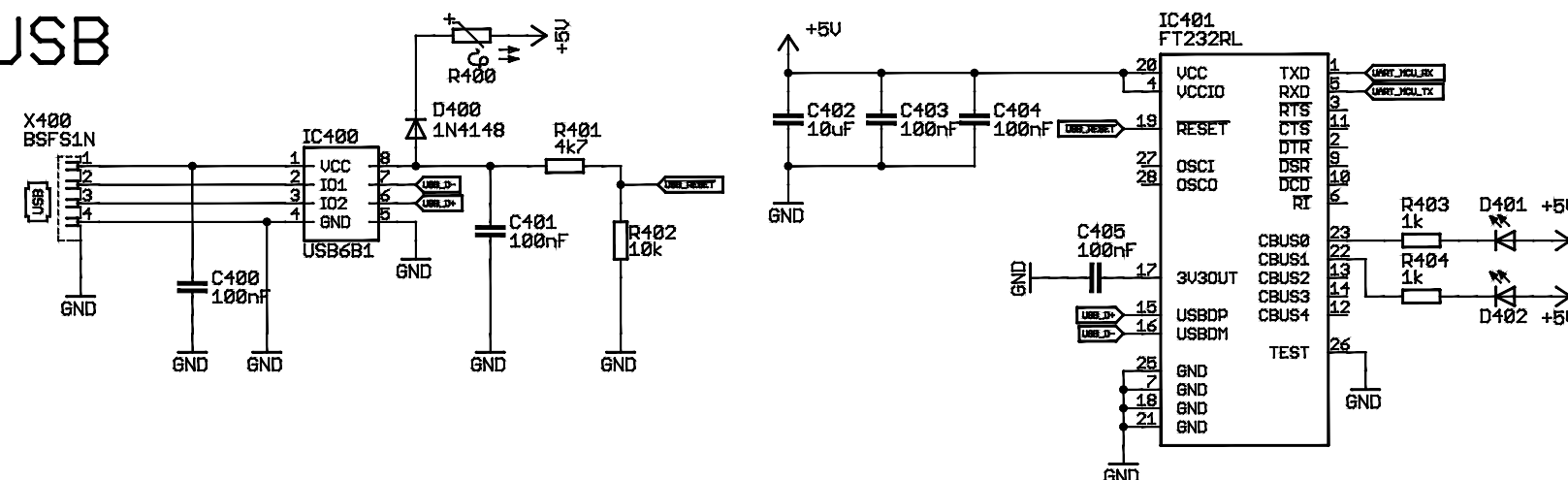
Power supply



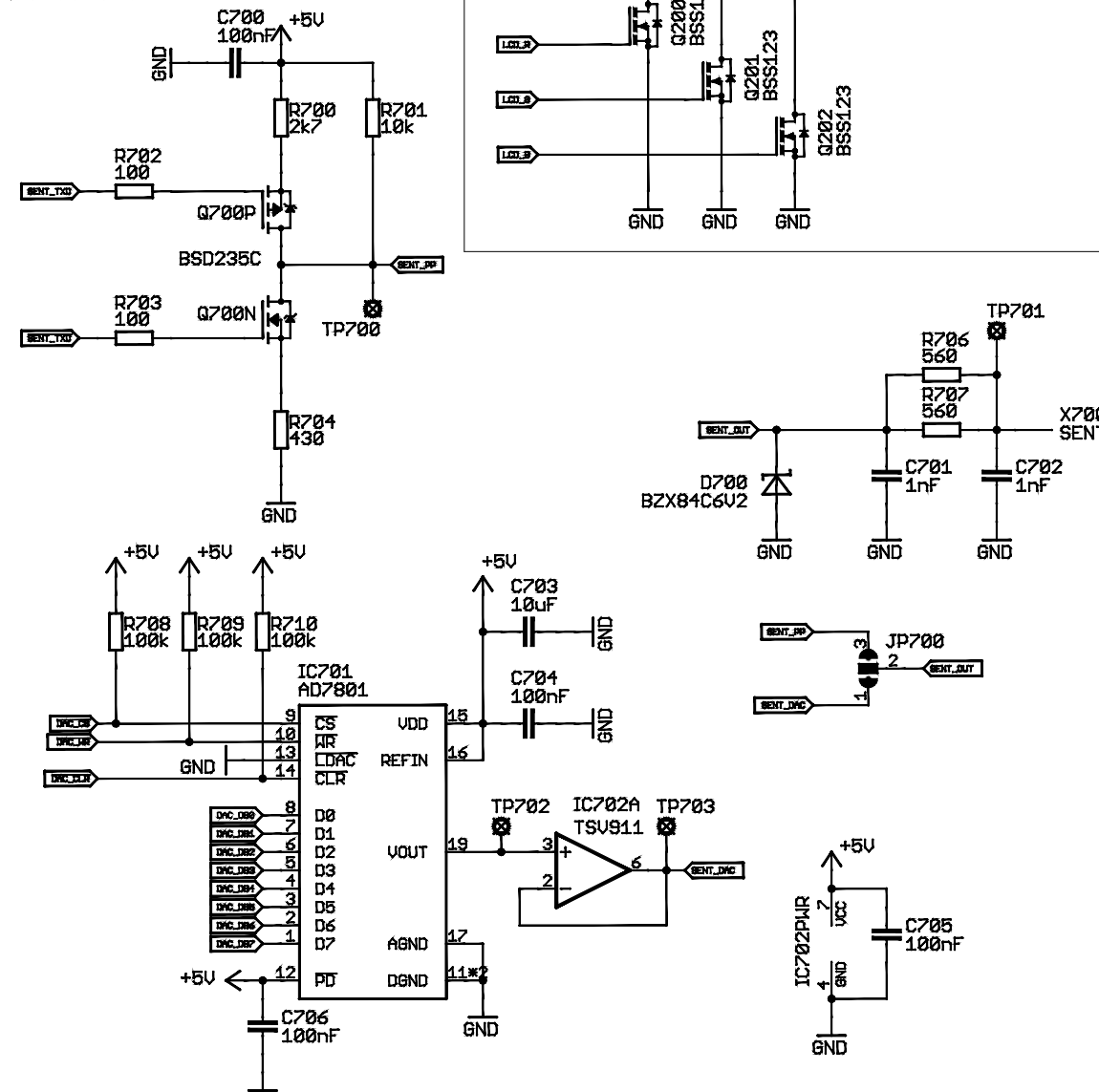
LCD



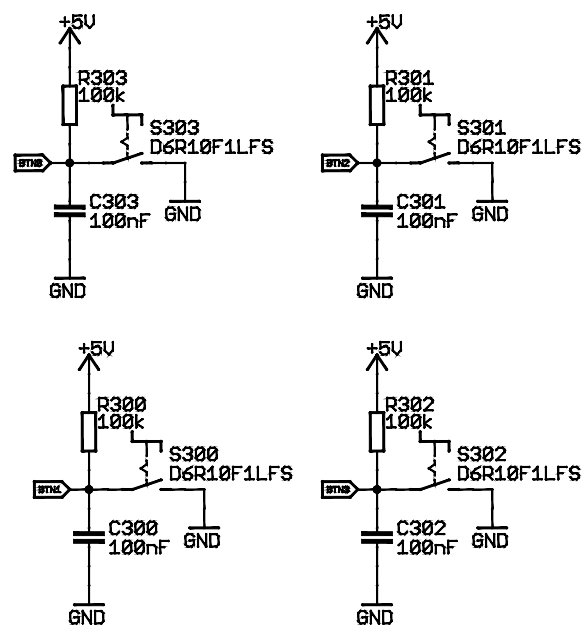
USB



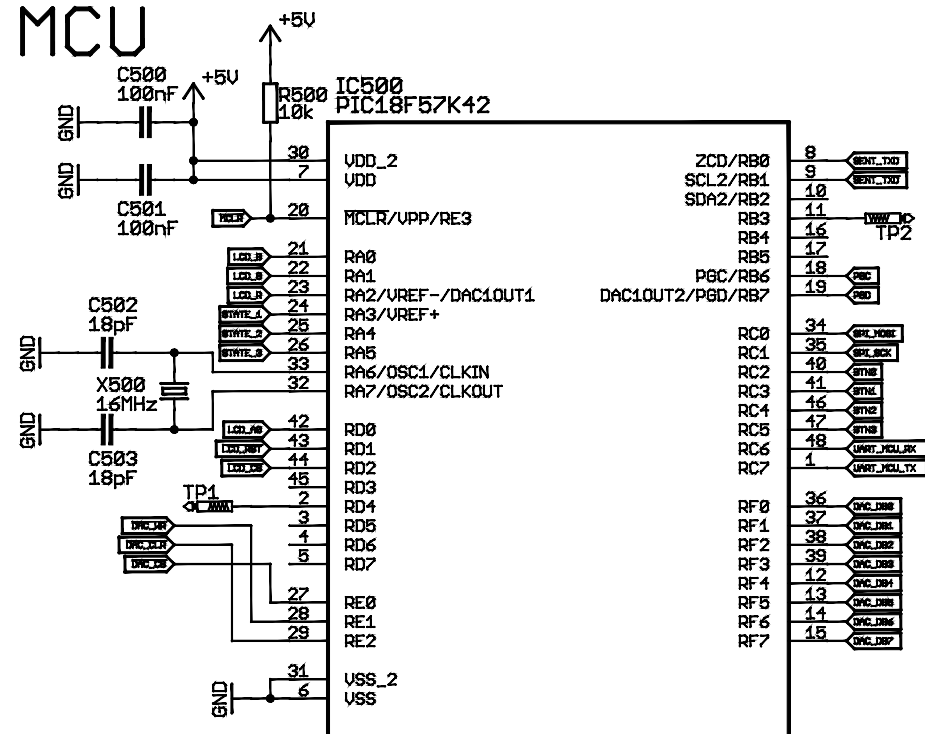
SENT



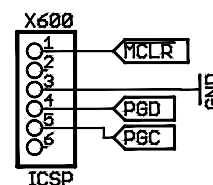
Buttons



MCU



ICSP



Designation	Date
Schematic design	14/01/2019
Designed by	Project name
TOMÁŠ SKOTNICA	HTS SENT Simulator
	Document number
	2019011400